



Fraunhofer Institut
Experimentelles
Software Engineering

SPEARMINT™ 7

User Manual

Authors:

Ulrike Becker-Kornstaedt
Fabio Bella
Jürgen Münch
Holger Neu
Alexis Ocampo
Jörg Zettel

IESE-Report No. 072.02/E
Version 1.1
January, 2003

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach
Sauerwiesen 6
D-67661 Kaiserslautern
Germany

Abstract

SPEARMINT™ is a process modeling tool that supports the efficient capture, and documentation of software development processes, as well as the maintenance and analysis of the resulting software process models. It supports graphical representations of process models, contains tailorable concepts for the definition and maintenance of role-oriented and personalized views, and provides interactive, direct-manipulative and hypermedia-integrated user interfaces. SPEARMINT™ integrates closely with a generator that can generate web-based handbooks as well as page-based handbooks of the process.

This user manual explains how to use the SPEARMINT™ tool to create software process models and generate Electronic Process Guides (EPGs). Additionally, it sketches the underlying concepts.

Keywords:

Software Process Model, Process Documentation, Process Definition, Software Process Views, Process Engineering, Process Modeling Tool, Process Improvement, SPEARMINT™, Electronic Process Guide, Process Handbook.

Table of Contents

1	Introduction	9
1.1	Scope of Tool and Document	9
1.2	Structure of Document	10
1.3	Literature	11
2	Basic Concepts	13
2.1	SPEARMINT™ Process Model Structure	13
2.2	SPEARMINT™ Entities	14
2.3	SPEARMINT™ Attributes	14
2.4	SPEARMINT™ View Concept	15
3	First Steps	18
3.1	Building a New Model	18
3.1.1	Creating a New Database	18
3.1.2	Creating a New Project	19
3.1.3	Creating a New Process Model	20
3.2	Setting Up the Project Attribute Profiles	21
3.3	Adding Basic Entities	23
3.3.1	Adding Activities	23
3.3.2	Adding Artifacts	24
3.3.3	Adding Roles and Tools	25
3.4	Working with a Product Flow View	25
3.4.1	Modeling a Product Flow Graph	25
3.4.2	Modeling a Product Flow Refinement Graph	30
3.5	Working with a Control Flow View	34
3.5.1	Modeling a Control Flow Graph	34
3.5.2	Modeling a Control Flow Refinement Graph	35
3.6	Working with a Role Involvement View	38
3.7	Working with a Resource Usage View	39
3.8	Working with a User-Defined Graphical View	40
3.9	Modifying Entities in the Properties View	43
3.10	Generating Process Guides	45
3.10.1	Generating an Electronic Process Guide	46
3.10.2	Generating Printable Process Documentation	48
3.10.3	Generating a Customized Process Guide	48
3.10.4	Generating a SPEARMINT™ 5 Electronic Process Guide	49
3.10.5	Transforming an XML File with XSLT	50
4	User Interface	51
4.1	General User Interface Structure	51

4.2	View Manager	52
4.3	Working Area	52
4.3.1	Menu, Toolbar and Status Bar	53
4.4	Popup Menus	57
4.4.1	Database Popup Menu	57
4.4.2	Project Popup Menu	58
4.4.3	Process Model Popup Menu	58
4.5	Popup Menus of the Predefined Views	60
4.5.1	Background Popup Menu	60
4.5.2	Node Popup Menu	61
4.6	Graphical Views	61
4.6.1	General Functions of Toolbars in Graphical Views	62
4.6.2	Background Popup Menu (only for Graphical View, Product Flow Graph and Control Flow Graph)	64
4.6.3	Edges Popup Menu	65
4.6.4	Grid and Snapping	66
4.6.5	Align and Distribute	69
5	Modelling with Views	72
5.1	Process Model Edition	72
5.1.1	Selection	72
5.1.2	Include and Exclude	73
5.1.3	Delete	74
5.1.4	Copy / Mark, Paste Copy, and Paste Object	74
5.1.5	Tree Structure Modification	75
5.2	Views	76
5.2.1	Activity View	76
5.2.2	Artifact View	76
5.2.3	Roles and Tools View	77
5.2.4	Product Flows View	78
5.2.5	Product Flow Graphs	78
5.2.6	Control Flow View	80
5.2.7	Control Flow Graphs	80
5.2.8	Role Involvement View	82
5.2.9	Resource Usage View	82
5.2.10	Graphical View (User-Defined View)	82
5.2.11	Hypertext View	83
5.2.12	Properties View	86
6	Consistency Assurance	88
6.1	Methodological Prerequisites	88
6.1.1	Process Metamodel	88
6.1.2	Consistency Rules	89
6.2	Working with Automated Consistency Assurance	95
6.2.1	Setting up Automated Consistency Assurance	95

6.2.2	Checking the Consistency of a Process Model	97
6.2.3	Showing and Correcting Inconsistencies on Selected Objects	98
7	Similarity Analysis	100
7.1	Introduction	100
7.1.1	Usage	100
7.2	Heuristics	106
7.3	Parameter setting	112
8	Electronic Process Guide (EPG) Generation	114
8.1	SPEARMINT™ 7 EPG generation:	114
8.2	SPEARMINT™ 5 EPG: Setting the Configuration	114
8.2.1	Changing the Settings for Graphical Views	114
8.2.2	Generating a SPEARMINT™ 5 EPG	117
8.2.3	Open a SPEARMINT™ 5 EPG	118
8.3	Structure of the EPG	118
9	Plugins	119
9.1	Calculate Basis Statistics	119
9.2	Calculate Size and Complexity	119
9.3	Find Undefined Attributes	120
9.4	Outdated Plugins	121
9.5	Writing Plugins	121

1 Introduction

Understanding software development processes is the basis for effective software process improvement. One key instrument to process understanding and communication is the explicit description of the processes and their representation by a software process model. However, real-world processes and their models are complex. To deal with this complexity, adequate tool support is necessary.

1.1 Scope of Tool and Document

A process modeling tool that was especially designed to deal with this type of complexity is the SPEARMINT™ /EPG tool set. The tool set consists of two components. The modeling environment SPEARMINT™ (Software Process Elicitation, Analysis, Review and Modeling in an Integrated Environment) assists a process engineer in creating and maintaining process models. The second component is a generator to create EPGs (= Electronic Process Guides), which are hypertext process handbooks that provide navigation facilities to support process performers during process execution.

An outstanding feature of SPEARMINT™ /EPG is that the tool allows for efficient modeling of different views of the process model. A view is a subset of the process model, focusing on specific aspects, for instance product flow, or processes relevant for a certain role (e. g., QualityReviewer, Designer). Different views may use different representation mechanisms. The concept of views is shown in Fig. 1.1. The views are kept consistent by SPEARMINT™. A view can be seen as a filter for information: Using different views on the process model, a Process Engineer can focus on exactly those aspects of the process that are relevant for his current task, while information that is not needed for this task is not shown. Thus, the complexity of process models can be largely reduced.

This document describes the usage of the SPEARMINT™ /EPG tool. The report assumes that the reader is familiar with fundamental software process modeling concepts.

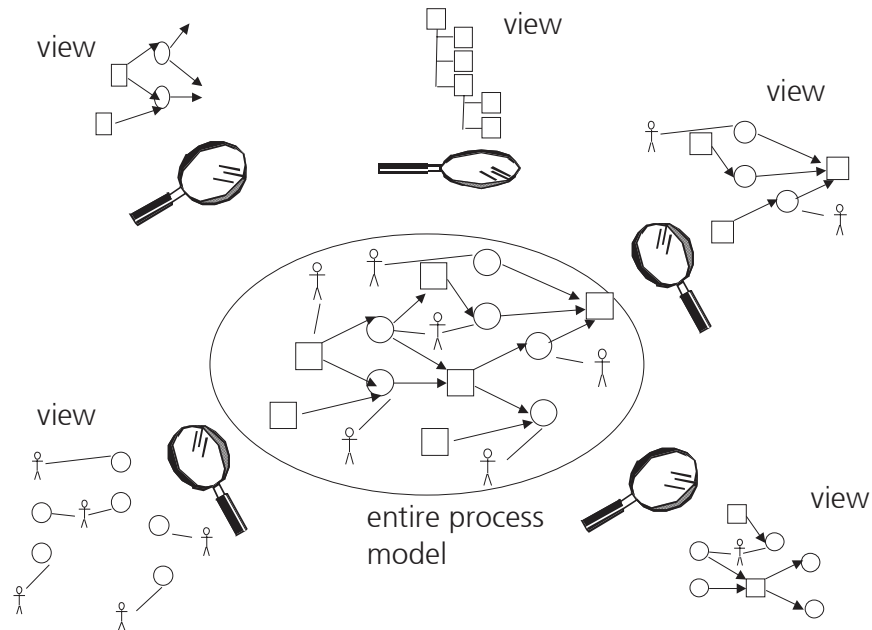


Figure 1-1 Complexity reduction with the SPEARMINT™ view concept

1.2 Structure of Document

This document is structured as follows. The next chapter introduces the basic concepts of SPEARMINT™, such as activities, artifacts, roles, tools, and their relations. Additionally, the structure of a SPEARMINT™ database, and the view concept implemented in the SPEARMINT™ tool are described.

Chapter 3 is a tutorial guide taking the user through his first steps of working in SPEARMINT™. This section can also function as a quick reference guide to basic questions such as how to create a new product flow graph or how to generate an EPG, etc.

Chapter 4 introduces the SPEARMINT™ user interface. The basic functionality is explained and the user interfaces of different standard views are introduced. The functionality of the various SPEARMINT™ menus are explained, i.e., the various context and background menus.

Chapter 5 describes the different views of SPEARMINT™ and how to edit the process model. It explains the special focus of each predefined view in SPEARMINT™ and what it displays. It also gives an account of the user-defined views.

Chapter 6 explains the methodological consistency rules of SPEARMINT™ operates and shows some examples.

Chapter 7 describes the functionality of the 'Similarity analysis' tool in SPEARMINT™.

Chapter 8 describes the usage of the generator for the EPG. It also describes how to change the configuration to fit the user's needs.

Chapter 9 describes the Plug-ins, which have been developed for SPEARMINT™ and their functionality.

1.3 Literature

The following literature provides further information about SPEARMINT™ /EPG from a research-oriented point of view.

Richard Webby, Ulrike Becker. "Towards a Logical Schema Integrating Software Process Modelling and Software Measurement." In *Process Modelling and Empirical Studies of Software Evolution. ICSE'97 Workshop*, pages 84–88, 1997.

Marc I. Kellner, Ulrike Becker-Kornstaedt, William E. Riddle, Jennifer Tomal, Martin Verlage. "Process Guides. Effective Guidance for Process Participants." In *Proceedings of the Fifth International Conference on the Software Process, ICSP'98*, pages 11–25, 1998.

Ulrike Becker-Kornstaedt, Dirk Hamann, Ralf Kempkens, Peter Rösch, Martin Verlage, Richard Webby, and Jörg Zettel. "Support for the process engineer: The SPEARMINT approach to software process definition and process guidance." In Matthias Jarke and Andreas Oberweis, editors, *Advanced Information Systems Engineering: 11th International Conference, CAiSE'99, Heidelberg, Germany, June 14–18, 1999, Proceedings*, number 1626 in Lecture Notes in Computer Science, pages 119–133. Springer-Verlag, 1999.

Ulrike Becker-Kornstaedt, Holger Neu, and Gunther Hirche. Software Process Technology Transfer: "Using a Formal Process Notation to Capture a Software Process in Industry". In Vincenzo Ambriola, editor, *Software Process Technology: 8th European Workshop, EWSPT 2001, Witten, Germany, June 19–21, 2001, Proceedings*, number 2077 in Lecture Notes on Computer Science, pages 63–76, Springer 2001.

Louise Scott, Lucila Carvalho, Ross Jeffery, and John D'Ambra. "An evaluation of the SPEARMINT approach to software process modelling". In Vincenzo Ambriola, editor, *Software Process Technology: 8th European Workshop, EWSPT 2001, Witten, Germany, June 19–21, 2001, Proceedings*, number 2077 in Lecture Notes on Computer Science, pages 77–89. Springer, 2001.

Louise Scott, Lucila Carvalho, Ross Jeffery, John D'Ambra, and Ulrike Becker-Kornstaedt. "Understanding the use of an electronic process guide." In *Information and Software Technology*, 44(10):601–616, July 2002.

Martin Verlage. "An Approach for Capturing Large Software Development Processes by Integration of Views Modelled Independently". In *Proceedings of the 10th International Conference on Software Engineering and Knowledge Engineering SEKE'98*, pages 227-235, 1998.





2 Basic Concepts

2.1 SPEARMINT™ Process Model Structure

SPEARMINT™ process *models* can be associated with several *views*. These views are the visual representations of parts of the process model. *Process Elements* in the views refer to the process elements in the model. Therefore, any change to a process element in some view causes a change in the process model, and becomes immediately visible in any other view in which this process element is visible. SPEARMINT™ offers predefined view types as well as views that can be defined by the user.

Multiple process models can be aggregated in a *project* and a set of projects can be stored in a *SPEARMINT™ database* (having the extension .spm).

SPEARMINT™ uses the following symbols:

- SPEARMINT™ file: 
- Project: 
- Process model: 
- View:  (predefined view)
 (user-defined graphical view)





Database	Project 1	Process Model 1.1	View 1.1.1
			View 1.1.2
			...
		Process Model 1.2	View 1.2.1
			View 1.2.2
			...
	Project 2	Process Model 2.1	View 2.1.1
			View 2.1.2
			...
		Process Model 2.2	View 2.2.1
			View 2.2.2
			...

Figure 2.1 Spearmint database structure

The structure of a SPEARMINT™ database:

2.2 SPEARMINT™ Entities

SPEARMINT™ uses a graphical notation to document processes. For the notation, icons were selected that are intuitively understandable. SPEARMINT™ has the following four *entity* types:

Entity	Icon	View that can contain the entity
• Activities		activities view, product flow refinement, product flow graph, control flow refinement, control flow graph, user-defined graphical view
• Artifacts		artifacts view, product flow graph, user-defined graphical view
• Roles		roles view, user-defined graphical view
• Tools		tools view, user-defined graphical view

2.3 SPEARMINT™ Attributes

The characteristics of SPEARMINT™ entities can be described using *attributes* (e.g., duration of a design activity, number of defects of a code document, experience level of a tester). SPEARMINT™ provides *default attributes* as shown in Fig 2.2

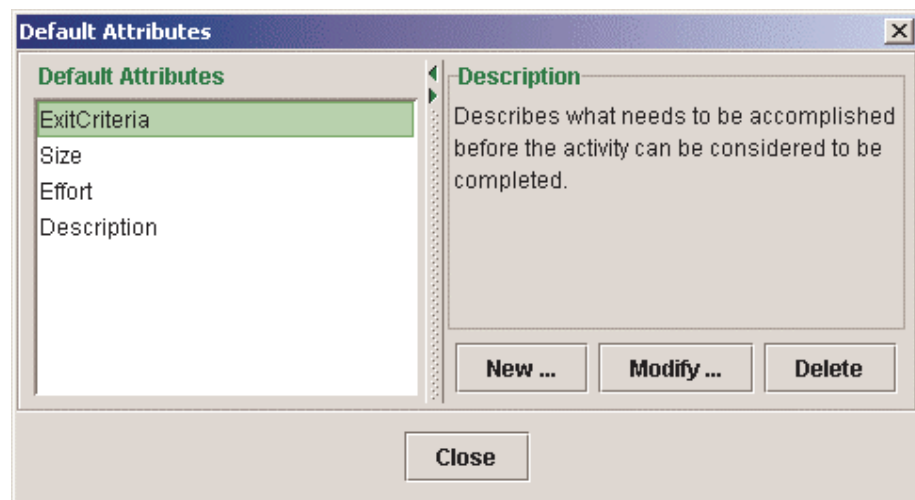


Figure 2.2

Default attributes

The importance of the default attributes is their generality. The attributes defined here will be attached automatically to each entity created afterwards.

An individual definition of your *own attributes* is allowed by SPEARMINT™. Each attribute has a name, a description, a value and a content type. Attributes are displayed in the properties view of the model.

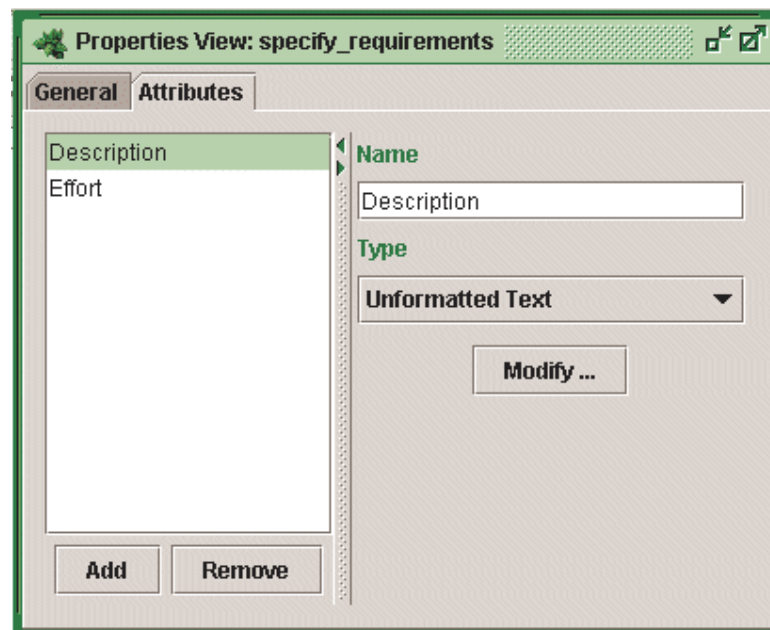


Fig 2.3

Attributes defined by the user

2.4 SPEARMINT™ View Concept

SPEARMINT™ provides different view types for accessing and editing the information in your process model. Views usually show only a part of the whole process model. They ensure that you see only the information you need for your immediate task.

In SPEARMINT™ there are eight predefined views for each process model. These show all the information *of a particular kind* and their content is automatically determined according to what they show:

- | | |
|-------------------|--|
| • Activities View | lists activities only and allows a display of their hierarchical structure |
| • Artifacts View | lists artifacts only and allows a display of their hierarchical structure |
| • Roles View | lists roles only |
| • Tools View | lists tools only |

- Product Flow View lists the *product flow graphs* (which model the product flow among activities and artifacts) and activities hierarchically
- Control Flow View lists the *control flow graphs* (which model the control flow among activities) and activities hierarchically
- Role Involvement View lists roles and their involvement in activities
- Resource Usage View lists resources, i.e., tools, and their usage in activities

The user can define other views. In this case, the user can choose which information is displayed in the view:

- Graphical view for modeling resource allocation relationships that specify roles and tools involved in a process by assigning them to specific activities (see figure 2.4)

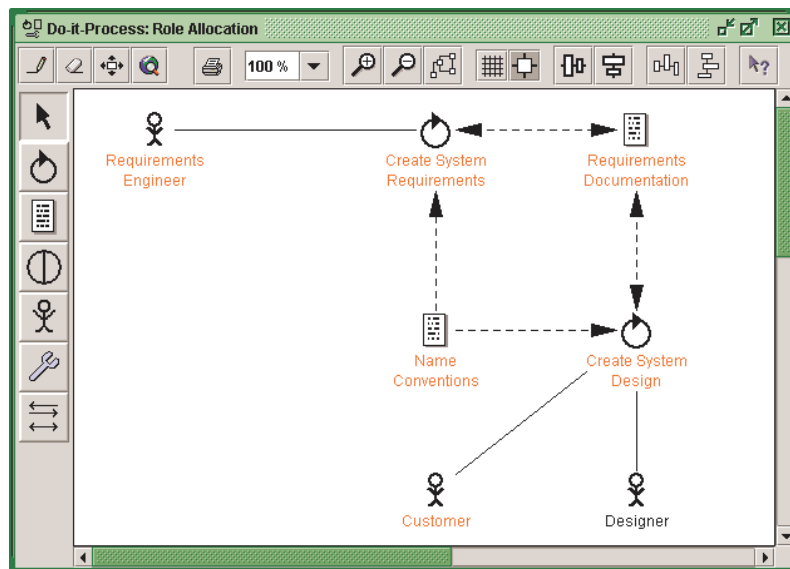


Fig 2.4

User-defined graphical view

- Hypertext View for viewing the model in a browser by loading the URL in a web browser (see figure 2.5)

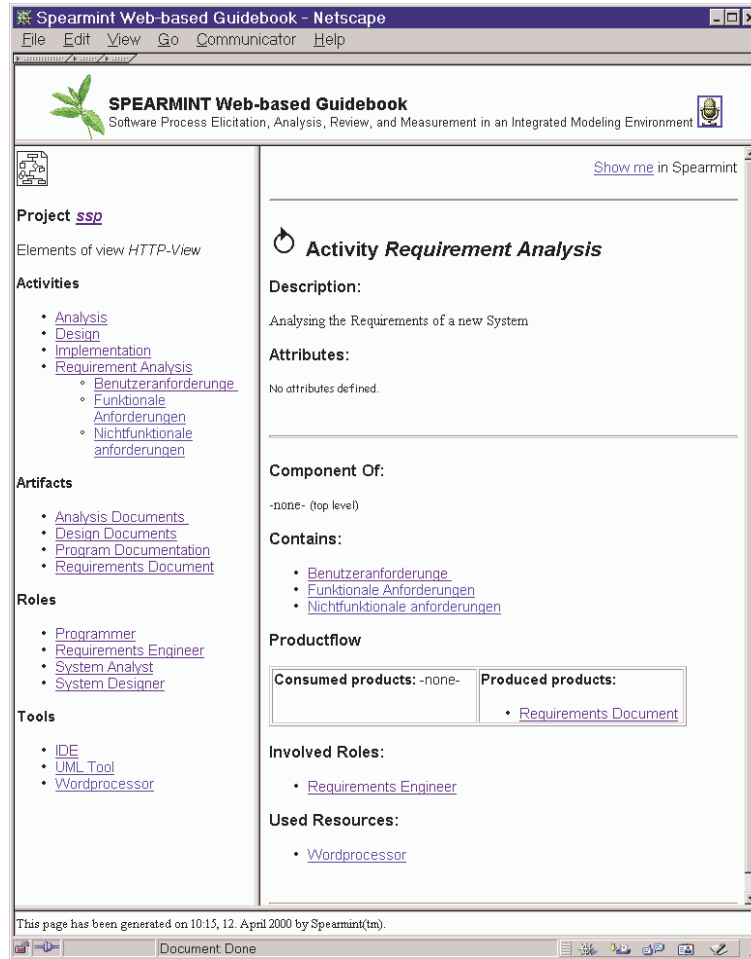


Fig 2.5

Hypertext view

- Properties View for giving additional information about a specific entity, e.g., defining attributes (see Fig 2.3).

In either case, the information in any view is always consistent with the information in the process model. If some information is changed in one view (e.g., changing a name, deleting an activity), the change is immediately reflected in all views that show this element.

3 First Steps

3.1 Building a New Model

In this section, an example process model will be created from scratch. The example can be used as an introductory tutorial. Guidance on how to model a process with the different predefined and user-defined views will be given.

3.1.1 Creating a New Database

- SPEARMINT™ will automatically load the database that was opened last when started.
- Close this database with the main file menu by choosing the option 'Close'.

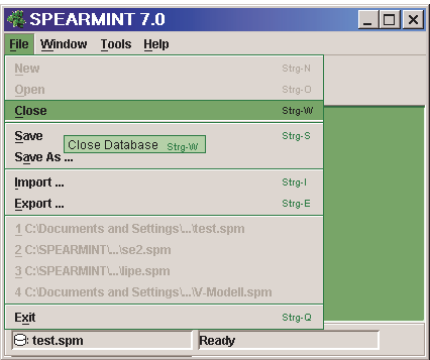


Figure 3.1

Closing a database

- To create a new file, use the main file menu by choosing the option 'New', which will only be activated when no database is opened.

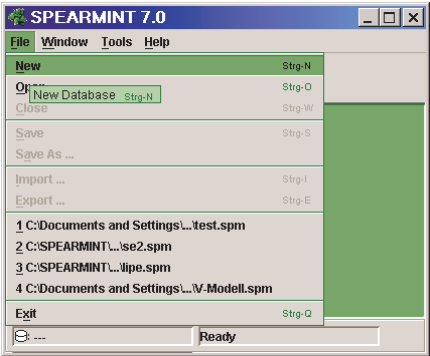


Figure 3.2

Opening a new database

- The dialog 'Create Database' will pop up. Chose a folder by double clicking, enter the file name 'IESE-Process-Repository' and click 'Create'.

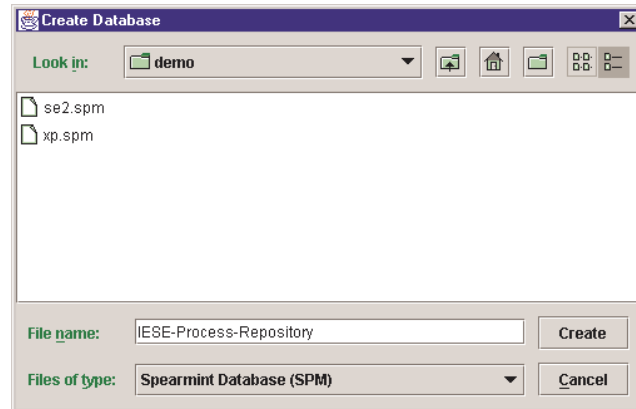


Figure 3.3

Creating a new database

3.1.2 Creating a New Project

A new database has been created in SPEARMINT™, which will now be seen in the database explorer that is the left-hand panel of the main window. To create a new project

- Use the popup menu by clicking with the right mouse button on the database named 'IESE-Process-Repository.spm'
- Choose the option 'New Project'.
- Enter the project name 'Embedded Software Development'.

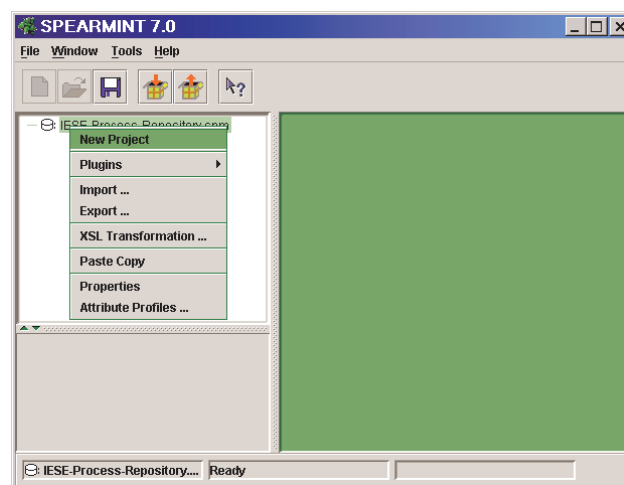


Figure 3.4

Creating a new project

3.1.3 Creating a New Process Model

A new process model is created in the same way.

- Use the popup menu of the project by clicking with the right mouse button on the project name 'Embedded Software Development' in the left-hand panel of the main window
- Choose the option 'New Process Model'.

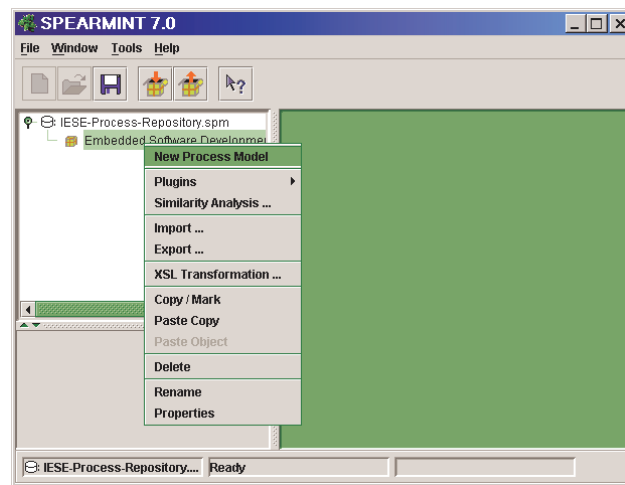


Figure 3.5

Creating a new process model

- Enter the process model name 'Do-it-Process' and hit the return key.
- The process model has now been created, and by clicking on the node beside the model name, the six predefined views can be seen in the database explorer.

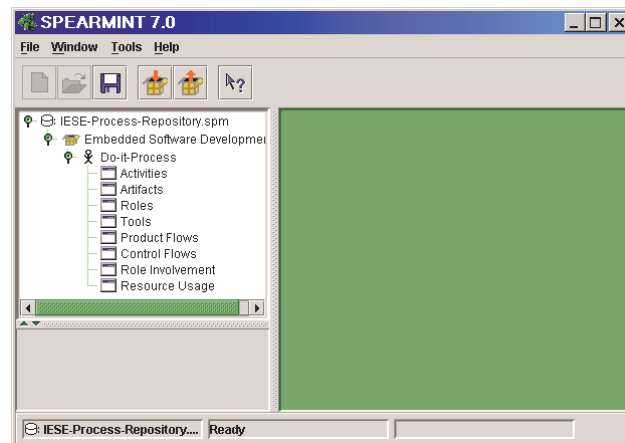


Figure 3.6

Structure of the SPEARMINT™ database

3.2 Setting Up the Project Attribute Profiles

The project IESE Process Repository could have several process models. Each of these process models could have a fixed set of attributes, which could be applied to all of its entities. These attributes are set using the option attribute profiles as follows

- Right click the IESE Process Repository node
- Select the attribute profiles option
- Click on new to create the set of attribute profiles which could be applied to one or several process models
- Name the attribute profiles as "Tutorial" and set its description as "Set of attributes for the model"
- Modify the attribute profiles for the Tutorial set by clicking the modify button
- Click on the button attributes to add or modify predefined attributes

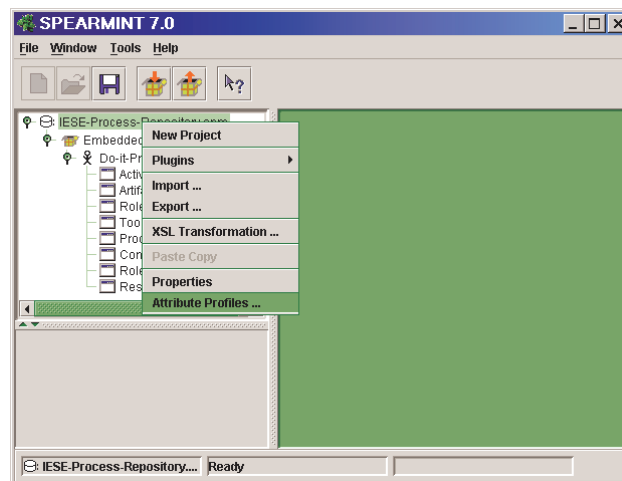


Fig 3.7

Set attribute profiles

- Click on the button attributes to add or modify predefined attributes
- Click on the button new to add the predefined attribute
- Set the following fields in the window "New default attribute"
 - **Name:** Enter "Description"
 - **Implicit feature of:** Select *any entity* which, means that roles, tools, activities, and artifacts will have at least a description

- **Type:** Select *Unformatted text* and enter "Short explanation of the entity". Unformatted text means that the value of the attribute will be edited as normal text. Other ways to edit values are: Formatted Text (HTML editor), Changes (Special table which introduces the version values of the attribute), and Choice (Option list with specific values).
- **Description:** Enter "Short explanation of the entity"

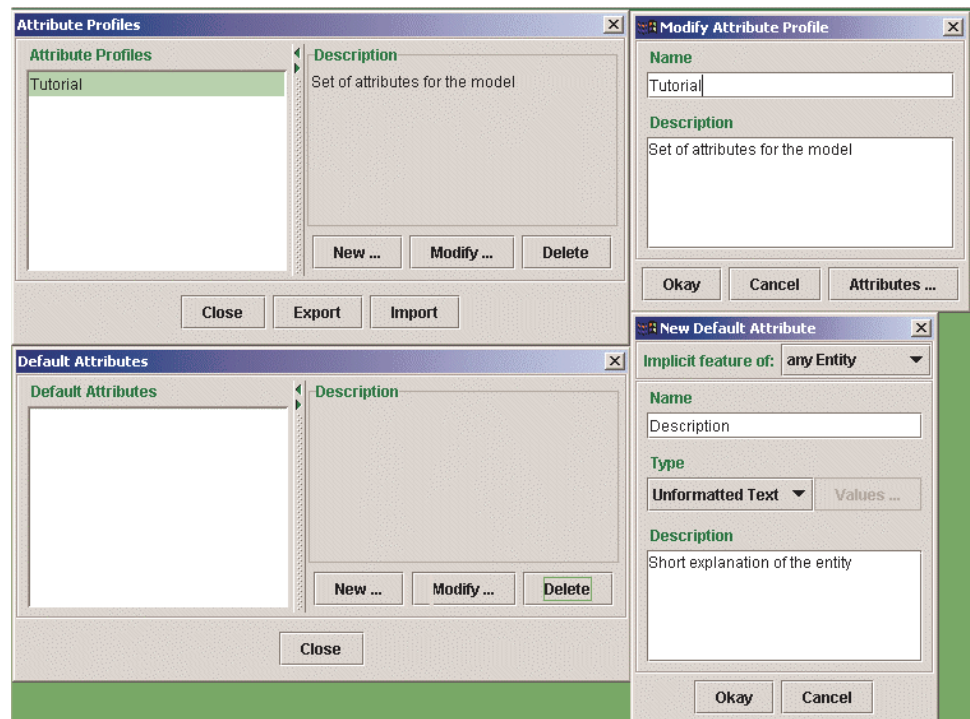


Fig 3.8

Step by step default attribute creation

- Select from the pop up menu of the process model the option "attribute profiles"
- Check Tutorial and activate the definition you have just set.

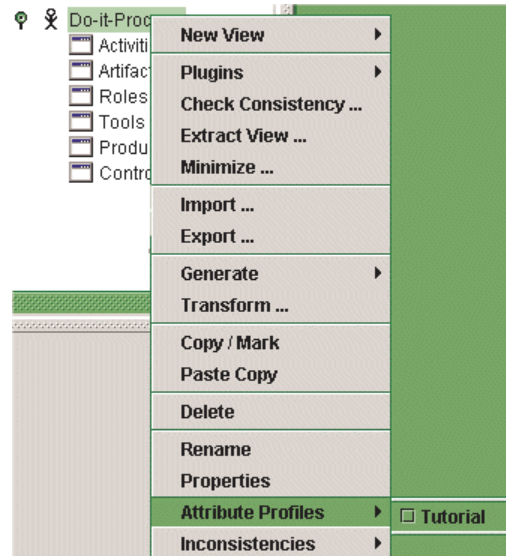


Fig 3.9 Activate the attribute profile

3.3 Adding Basic Entities

There is a specific view for each of the four SPEARMINT™ entities. New entities can be added in these views. There are other ways of adding new entities in the flow views, user-defined graphical views, and graphs.

3.3.1 Adding Activities

- Open the activities view window by double clicking 'Activities' in the database explorer.
- Now click with the right mouse button anywhere in this window and the background context menu will pop up.
- Choose the option 'New Activity'.

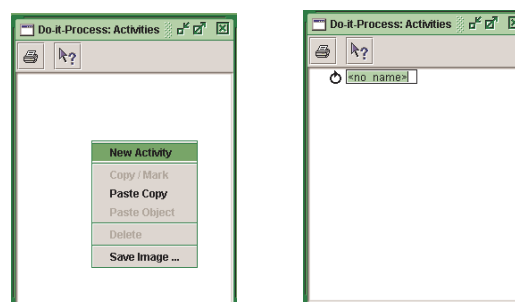


Figure 3.10 Creating a new activity

- A delayed double click allows to rename the activity.
- It is also possible to use the popup menu of the specific activity by right-clicking with the mouse on the activity for the context menu of this item and choosing the option 'Rename'.
- Create the following activities and note that they are automatically listed in alphabetical order:

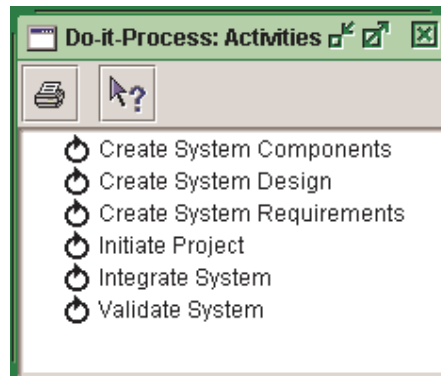


Figure 3.11

The activities view

3.3.2 Adding Artifacts

Artifacts can be created in the same way:

- Open the artifacts view window by double clicking 'Artifacts' in the view manager.
- Now click with the right mouse button anywhere in this window for the background context menu to pop up.
- Choose the option 'New Artifact'.

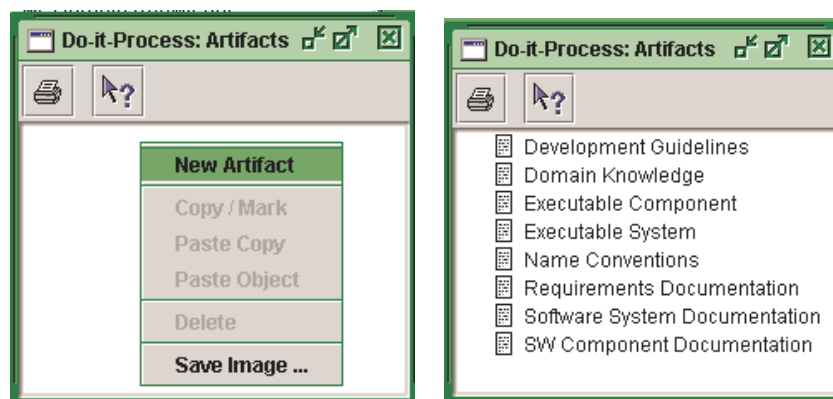


Figure 3.12

Creating new artifacts

- A delayed double click allows to rename the artifact.

- Again, it is also possible to use the popup menu of the specific artifact.
- Create the artifacts listed above.

3.3.3 Adding Roles and Tools

The Roles and Tools are non-hierarchical views merely listing the objects.

- Open the roles (tools) view window by double clicking 'Roles' ('Tools') in the view manager.
- Now click with the right mouse button anywhere in this window for the background context menu to pop up.
- Choose the option 'New Role' ('New Tool'):

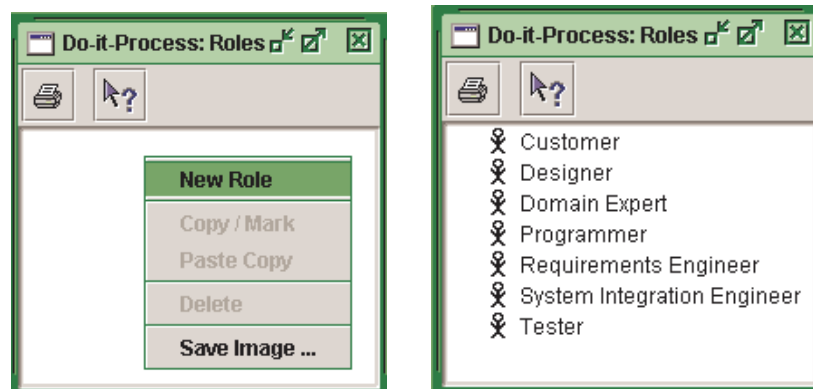


Figure 3.13

Creating new roles

- A delayed double click allows to rename the role (tool).
- Again, it is also possible to use the popup menu of the specific role (tool).
- Try to create the roles listed above.

3.4 Working with a Product Flow View

The product flow view hierarchically lists the product flow graphs. The product flow graphs model the product flow between activities and artifacts.

3.4.1 Modeling a Product Flow Graph

- Open the product flows window by double clicking 'Product Flows' in the database explorer
- Notice that the activities have been included into this view automatically.
- Create a new product flow graph by using the background popup menu and choosing the option 'New Product Flow Graph':

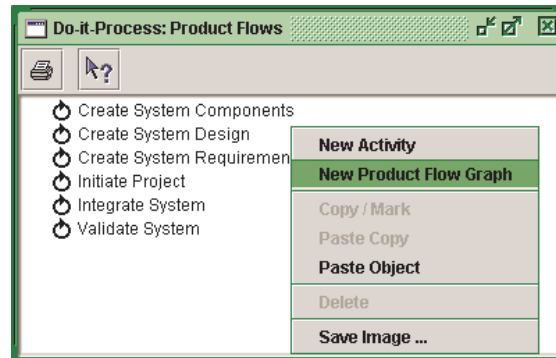


Figure 3.14

Creating a new product flow graph

- A delayed double click allows to rename the graph 'General View':

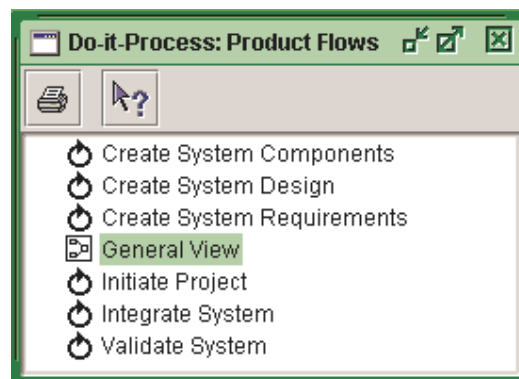


Figure 3.15

A product flow graph in the product flow view

In order to include the activities into the graph they have to be moved:

- Mark the activities by holding the key Strg or Shift while clicking on each activity.
- Right click the marked activities for the context menu to pop up and choose the option 'Copy/Mark':

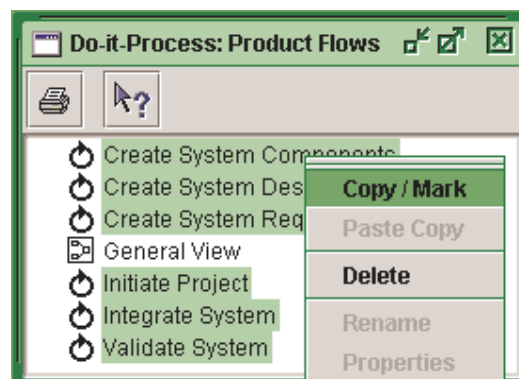


Figure 3.16

Copy / Mark option on the context popup menu

- Now mark the product flow graph
- Bring up its context menu by using the right mouse button and choose the option 'Paste Object'.

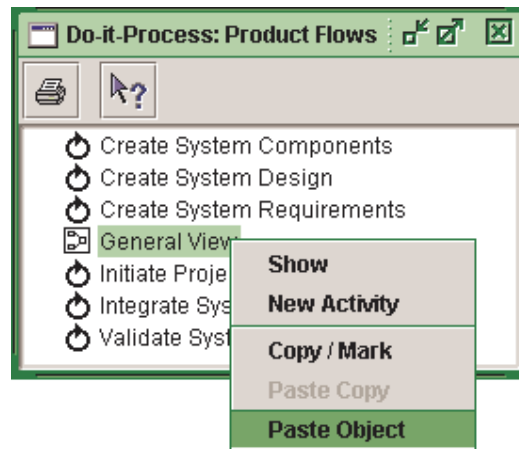


Figure 3.17

Past Object option on the context popup menu

- The activities will now be included in the graph (which is also reflected by the hierarchy in the product flow view).
- Double click the product flow graph 'General View' to open it. The activities can easily be positioned by dragging them with the mouse:

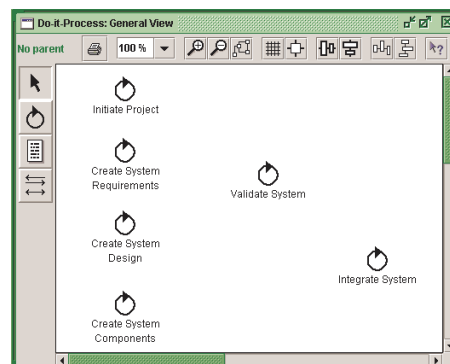


Figure 3.18

Activities in a product flow graph

Something to remember: If you have selected 'Paste Copy' instead of 'Paste Object' then you have created a copy of the objects. If you want to delete these copies from the model, open the activities view, select them and delete them.

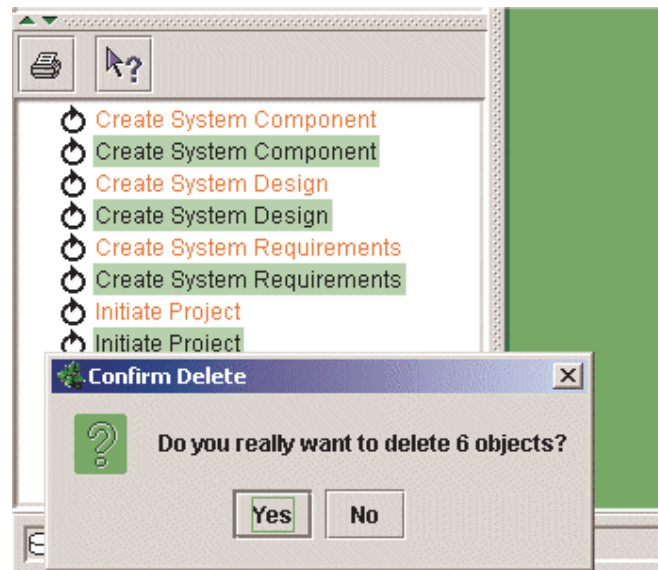


Fig 3.19

Duplicated activities

- 'Copy/Mark' the artifacts in the artifacts view
- Include them with 'Paste Object' into the graph:

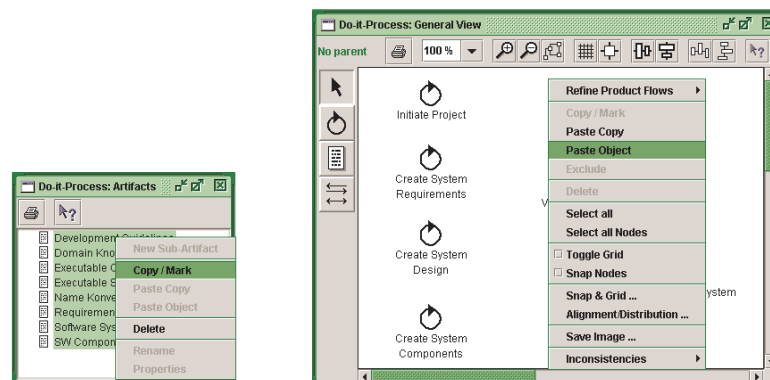


Figure 3.20

Including entities into the product flow graph

Something to remember: If you have selected 'Paste Copy' instead of 'Paste Object' then you have to go to the artifacts view and delete the duplicated artifacts.

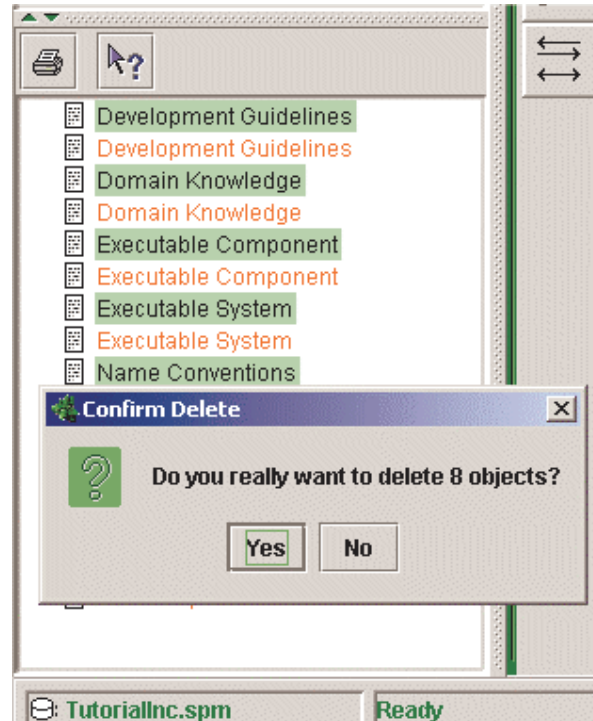


Fig 3.21

Duplicated artifact



- To work on any object in the graph, the 'Select Objects' button of the vertical tool bar has to be activated. Arrange the objects in the graph by dragging them into position with the mouse.
- Now activate the button 'Link Objects' in the vertical tool bar
- Add links by clicking an object, holding the left mouse button and moving the mouse to the object that you wish to link it to.
- To create a modifying link, repeat this procedure in both directions. The graph should look similar to the following:

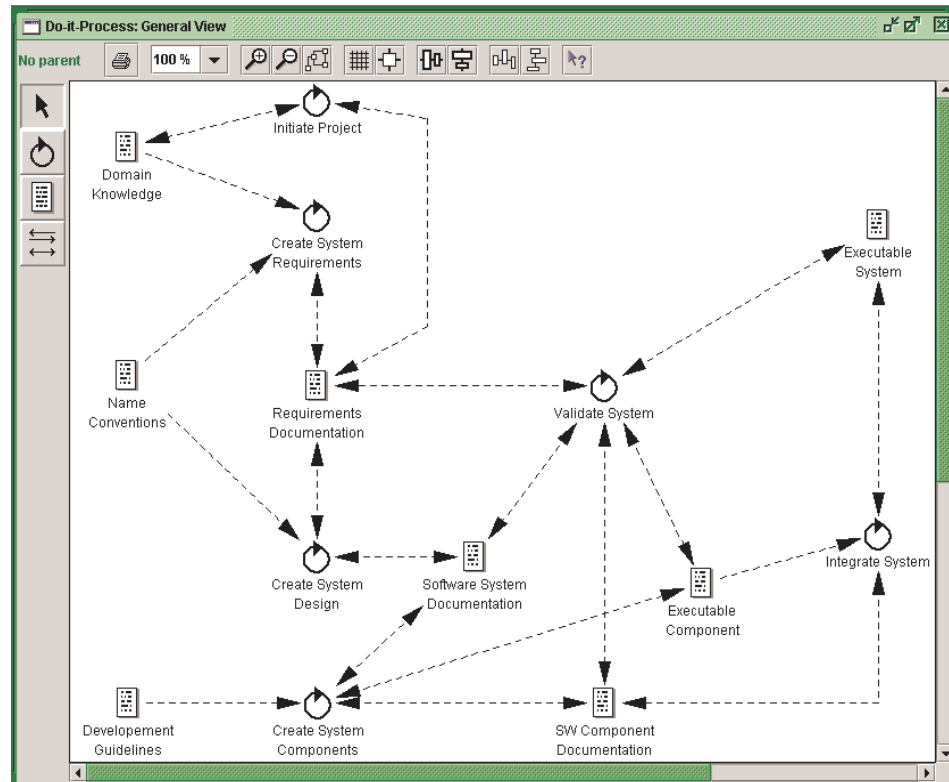


Figure 3.22 The completed product flow graph.

3.4.2 Modeling a Product Flow Refinement Graph

To create a refined view of any specific activity

- Go to the product flow view
- Right click that activity for the context menu to pop up
- Choose the option 'New Product Flow Refinement'. Here the activity 'Create System Requirements' is going to be refined:

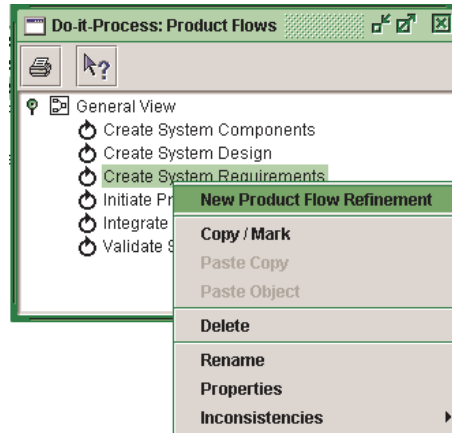


Figure 3.23

Creating a new product flow refinement

- The refinement graph of the activity will automatically be named after the activity.
- Open the still empty graph with a double click.
- Activate the 'Add Activity' button on the vertical tool bar of the graph to add new activities by clicking once on the graph for each new activity to be added.
- The new activities will automatically be shown in the hierarchical structure of the product flows view as subactivities:

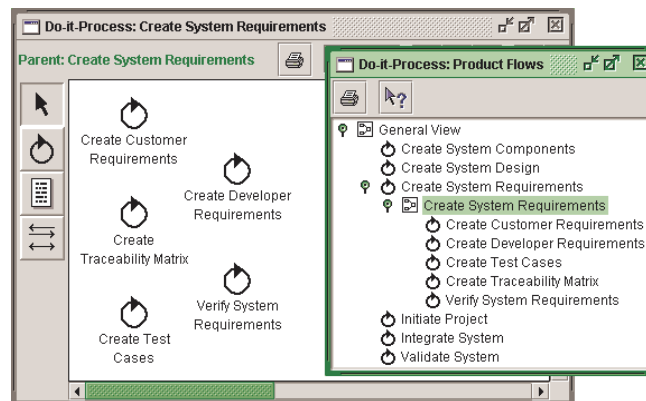


Figure 3.24

Product flow view and product flow refinement graph

The activities can be moved outside a Refined product flow graph as follows:

- Select the activity "Create Test Cases"
- Right click that activity for the context menu to pop up
- Select Copy/Mark

- Select the “General View” node in the Product Flow View
- Select Paste/Object
- The activity is taken outside the Refined product flow graph and outside the activity Create System Requirements as shown

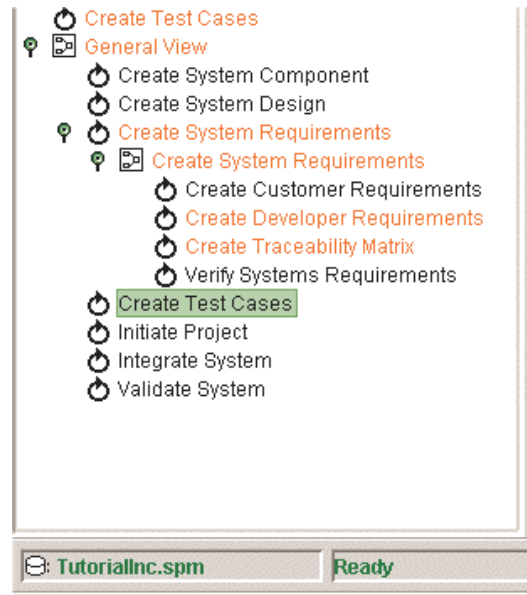


Fig 3.25

Activity has been moved outside the Product Flow Refined graph

- Move back the activity “Create Test Cases” into the Product Flow Refined graph “Create System Requirements”.

The product flow refinement graph of this activity will also include artifacts that are subartifacts of those already modeled. To add subartifacts

- Open the artifacts view
- Click the specific artifact with the right mouse button for the context menu to pop up and choose the option ‘New Subartifact’.
- Here the artifacts ‘Domain Knowledge’ and ‘Requirements Documentation’ have been substructured:

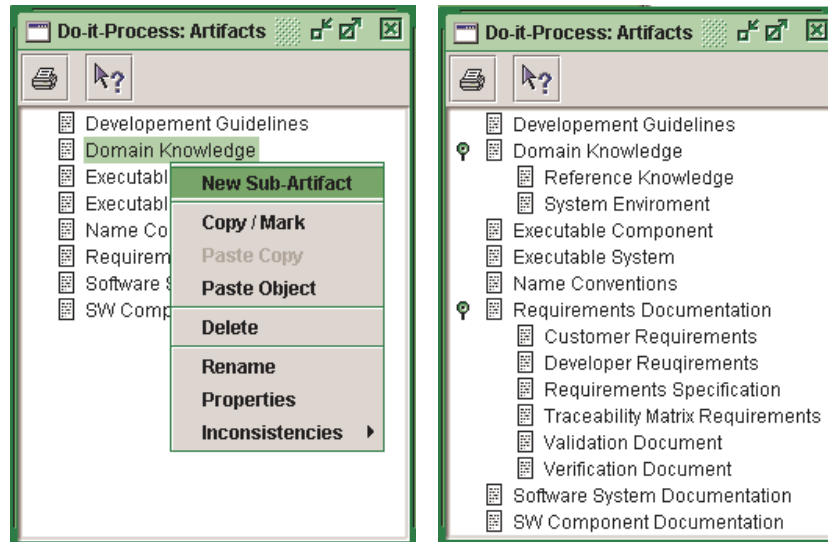


Figure 3.26

Creating Subartifacts

Use the option 'Copy/Mark' in the context menu of the artifacts to mark the newly added subartifacts as well as the artifact 'Name Conventions'. Use the option 'Paste Object' on the background popup menu of the target graph to include them into graph. (see Figure 3.16)

Arrange the objects and link them using the toolbar and the mouse. The product flow refinement of the activity 'Create System Requirements' is shown below:

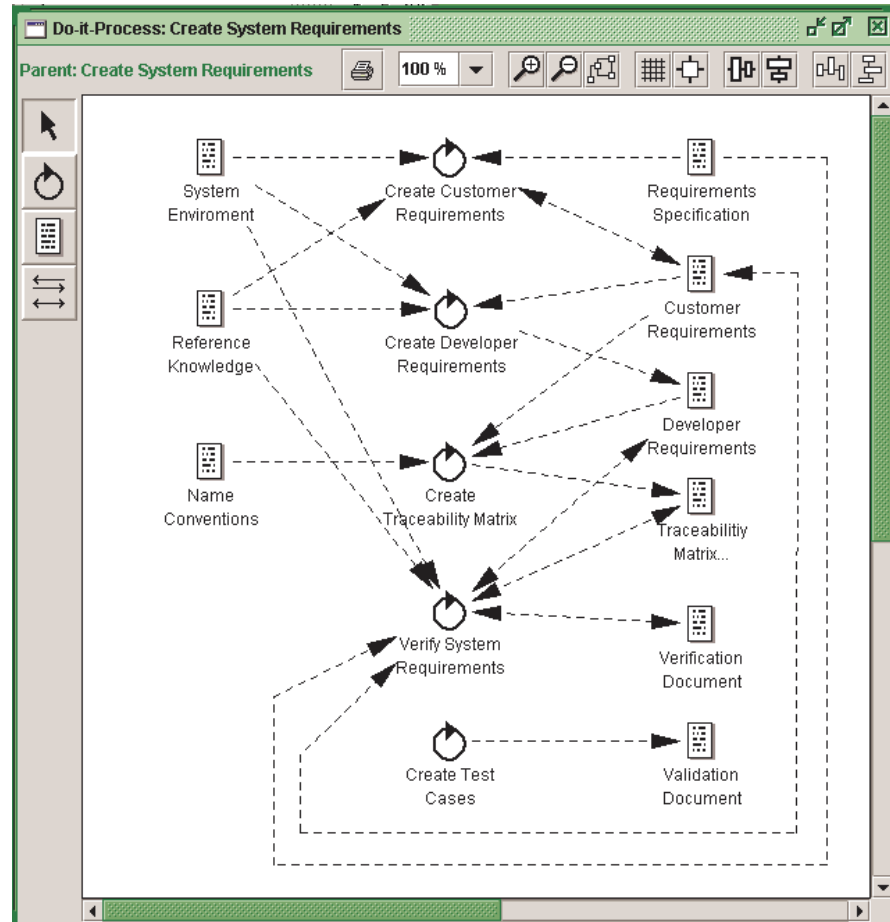


Figure 3.27 Product flow refinement graph of the activity 'Create System Requirements'

3.5 Working with a Control Flow View

The control flow view hierarchically lists the control flow graphs. The control flow graphs model the control flow among activities.

3.5.1 Modeling a Control Flow Graph

Opening the control flow view by double clicking 'Control Flows' in the database explorer, you will find that all activities have automatically been included in this view.

To create a general control flow graph

- Use the background popup menu of the control flow view, choose the option 'New Control Flow Graph'

- Name it 'General Control Flow Graph'.

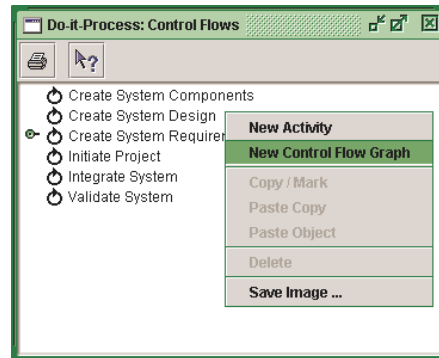


Figure 3.28

Background popup menu of the control flow view

- To include the activities into the graph, use the option 'Mark/Copy' on the context menu of the activities
- 'Paste Objects' on the context menu of the graph.

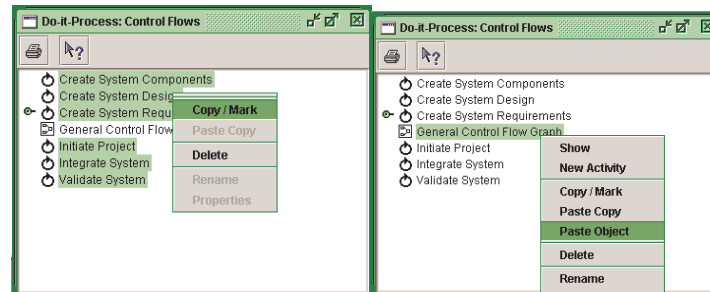


Figure 3.29

Moving entities within the control flow view

Now the control flow graph can be opened by double click and modeled.

3.5.2 Modeling a Control Flow Refinement Graph

After creation of the subactivities of the main activity 'Create System Requirements', a control flow refinement graph is automatically created into the hierarchical order of the control flow view including all subactivities. The refinement graphs are automatically named after their main activity:

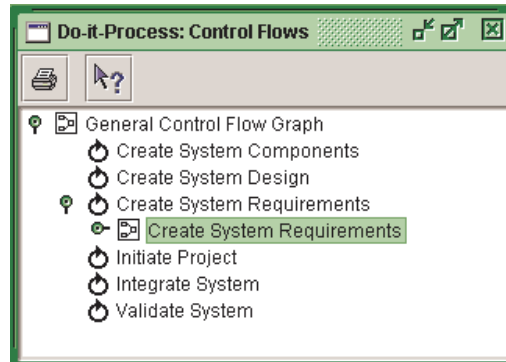


Figure 3.30

Control flow view

- Open the refinement graph 'Create System Requirements' by double clicking. The subactivities have been included into the graph automatically.
- Activate the 'Link Objects' button on the tool bar and link the objects

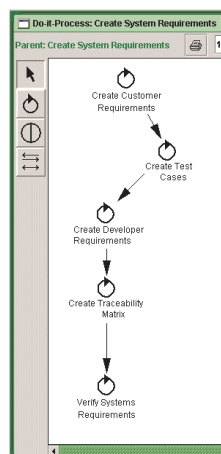


Fig 3.31

Basic control flow

The control flow can represent activities in parallel. This is done using the join/split states.



To link an activity to more than one other activity, join/split states have to be used to add a join/split state

- Activate the 'JoinSplitState' button on the vertical tool bar
- Click once in the working area of the graph to include a new state

- To change the status of the joint/split label, right click the label and choose the option 'Set Split' and then 'And'.
- Change the diagram to look now as follows:

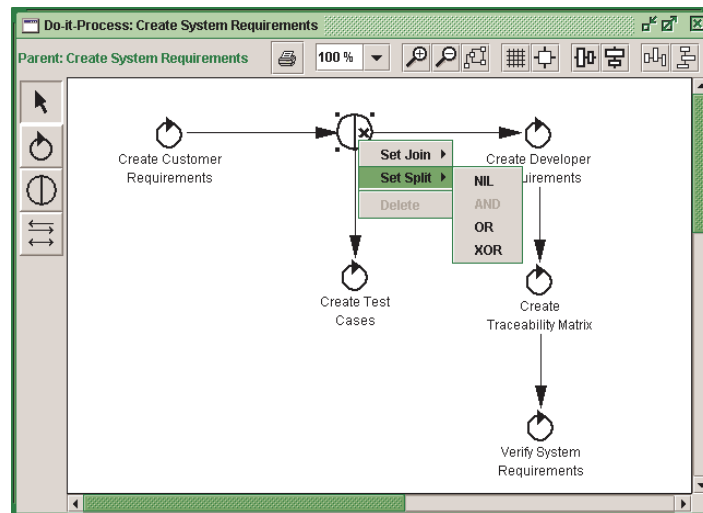


Figure 3.32

Control flow refinement graph

The Set split option represents a state, which is succeeded by more than one activity

The right half of the split/joint symbol is used here to indicate that the control flow of the activity 'Create System Requirements' is split.

To be more precise, the control flow divides itself in two branches. One goes to the activity 'Create Developer Requirements' and the other to the activity 'Create Test Cases'.

The AND option states that the activity 'Create Traceability Matrix' can not be done before the previous 'Create Test Cases' and 'Create Developer Requirements' are executed.

- Set a join option and change the diagram to look now as follows:

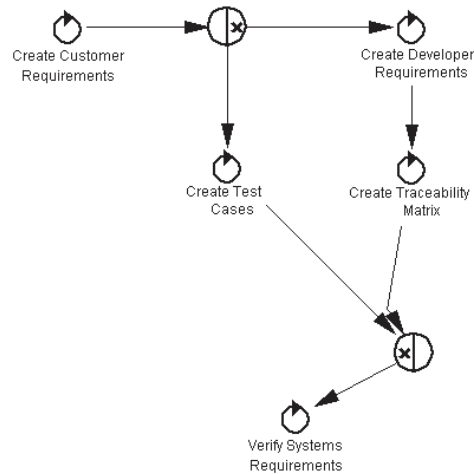


Fig 3.33

Control flow graph with split and join states

The Set join option represents a state, which is preceded by more than one activity.

Here, the control flow unifies itself into one branch. The AND option states that the activity 'Verify Systems Requirements' can be executed if and only if the activities 'Create Test Cases' and 'Create Traceability Matrix' were executed.

The following are the meanings for the AND, OR, XOR and NIL options.

- AND: It means that at least two activities have to be performed before or after the new state
- OR: It means that any activity of at least two activities could be performed before or after the new state.
- XOR: It means that exactly one out of a group of several activities could be performed before or after the new state.
- NIL: It means that no activity is involved.

3.6 Working with a Role Involvement View

The role involvement view lists the roles and their involvement in activities in tabular form.

- Open the role involvement view by double clicking 'Role Involvement' in the database explorer.
- Notice that roles are listed on the left side, whereas activities are listed at the top.

- Role names and activity names appear abbreviated. Move the mouse over a role to see the full role name in the tool tip.
- Create a new involvement relation by clicking in a table cell. The cell becomes checked, indicating that the role is involved in the activity.
- Delete the involvement relation by clicking in the table cell again. The cell becomes unchecked, indicating that the role is not involved in the activity.
- Create meaningful involvement relations between roles and activities as you like.

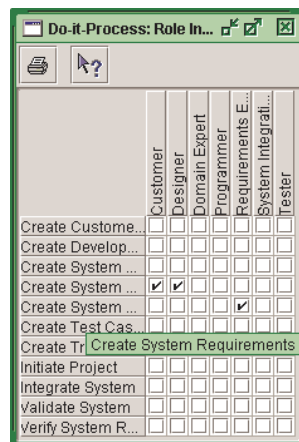


Figure 3.34

Role involvement view

3.7 Working with a Resource Usage View

The resource usage view lists the resources, i.e., tools, and their usage in activities in tabular form.

- Open the resource usage view by double clicking 'Resource Usage' in the database explorer.
- Notice that tools are listed on the left side, whereas activities are listed at the top.
- Tool names and activity names appear abbreviated. Move the mouse over a tool to see the full tool name in the tool tip.
- Create a new usage relation by clicking in a table cell. The cell becomes checked, indicating that the tool is used in the activity.
- Delete the usage relation by clicking in the table cell again. The cell becomes unchecked, indicating that the tool is not used in the activity.

- Create meaningful usage relations between tools and activities as you like.

3.8 Working with a User-Defined Graphical View

The graphical view is a user-defined view and thus the user can determine what the view is going to show.

- To create a new user-defined graphical view, use the process model popup menu in the database explorer and choose the option 'New View' and 'Process'.
- Name the new view 'Role Allocation':

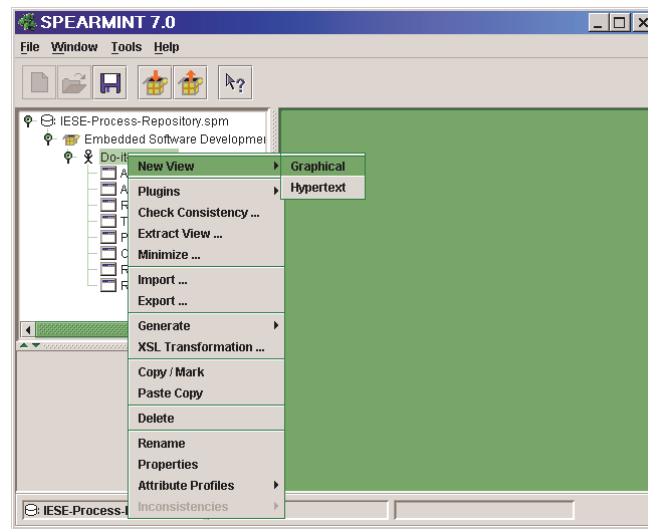


Figure 3.35

Process model popup menu



- Open the activities view and the new user-defined graphical view.
- Select the activities 'Create System Requirements' and 'Create System Design'
- Include them into the new view by using the 'Include selected' button on the horizontal tool bar:

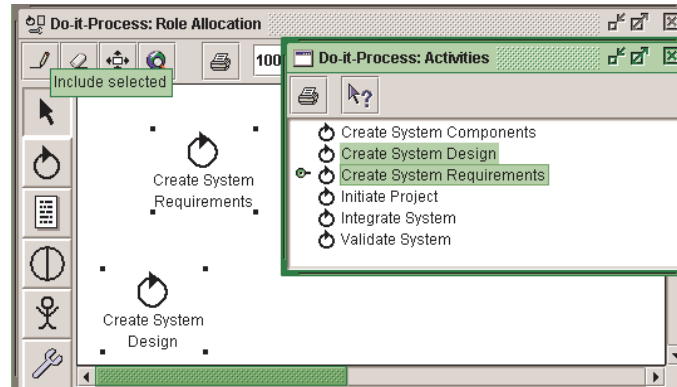


Figure 3.36

Including basic entities into a user-defined graphical view



- Select the activity 'Create Design system' and use the 'Include neighbors' button on the horizontal tool bar to include all entities this activity is linked to in other views (that is as long as the user-defined graphical view allows for their display).

Something to remember: The buttons on the horizontal tool bar 'Include' and 'Include neighbors' create a reference to the object in the model. If you delete one of the objects you have just included from the view, you erase it from the model. This generates an inconsistency.

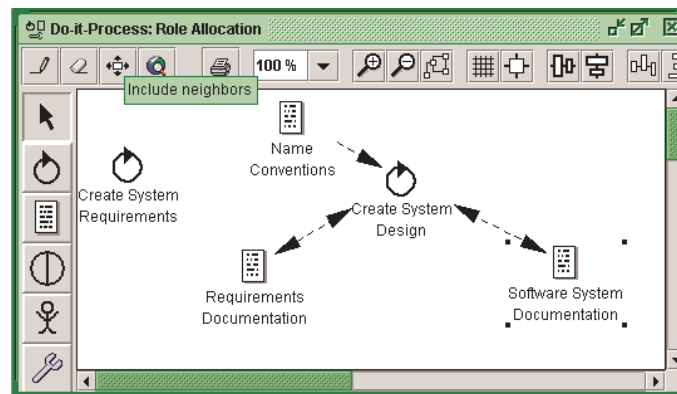


Figure 3.37

Including relationships of entities into a user-defined graphical view



- Mark the artifact 'Software System Documentation'
- Use the 'Exclude selected' button on the horizontal tool bar to exclude this entity from the graph.

In the same way, roles can be included into the graph

- Select them in the roles view and use the 'Include selected' button on the horizontal tool bar:

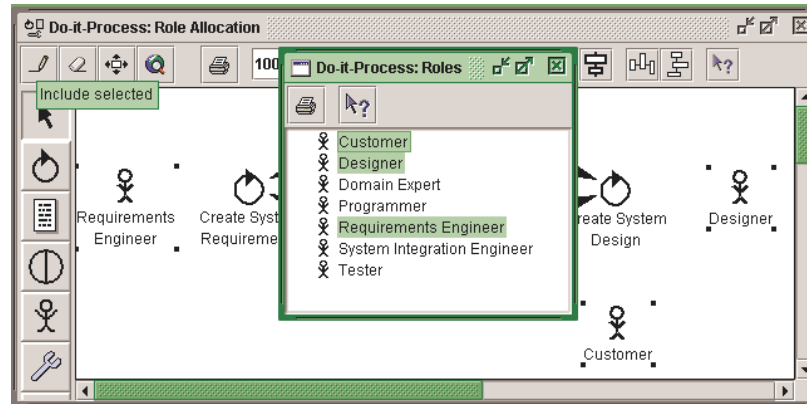


Figure 3.38

Including roles from the roles view into the user-defined graphical view.

- Now activate the 'Link Objects' button on the vertical tool bar
- Link the roles to the activities to complete the view:

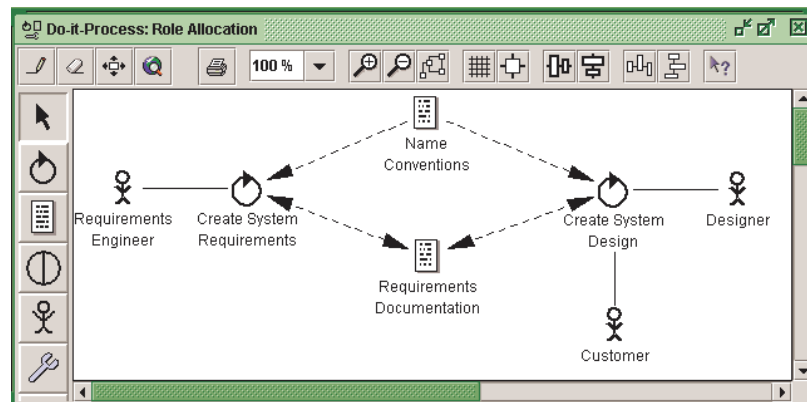


Figure 3.39

A completed user-defined graphical view.

Something to remember: The option 'Include neighbors', includes in the graph the objects and links which are related in some way with the selected object. If there are activities or artifacts, which don't have a link connecting them to the selected object, then those are not from the same level. It is recommended to use the 'Exclude' option to remove these objects from the view in order to obtain a representation with elements from the same level.

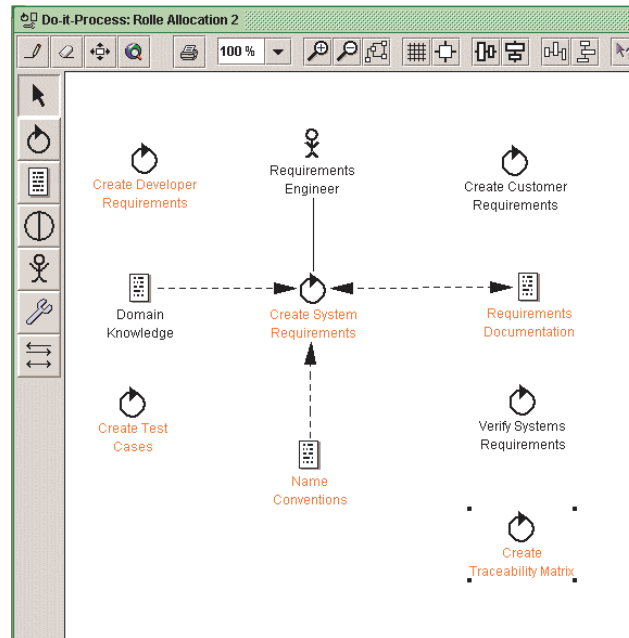


Fig 3.40

Different level activities are shown in the same view.

The above Figure 3.31 shows an example of this situation. The view contains activities and artifacts, which don't belong to the same level. The activities are from the main level and the artifacts belong to the activity Create Systems Requirements level.

3.9 Modifying Entities in the Properties View

The properties of an entity can be viewed in the properties view. In order to view the properties for the artifact 'Requirements Documentation'

- Go to any view containing this artifact, e.g., the artifacts view.
- Click on the entity with the right mouse button
- Choose the option 'Properties' from the popup menu:

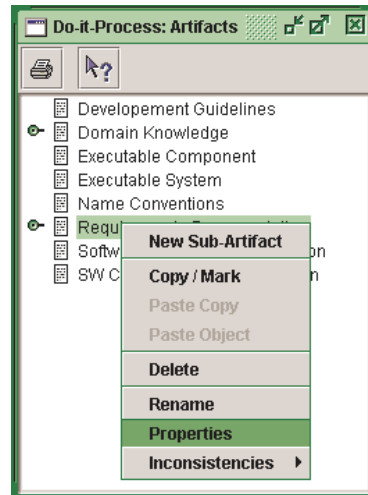


Figure 3.41

Properties option on the context menu

- In the properties view there are two panels. Choose the general tab and enter the description for the entity.

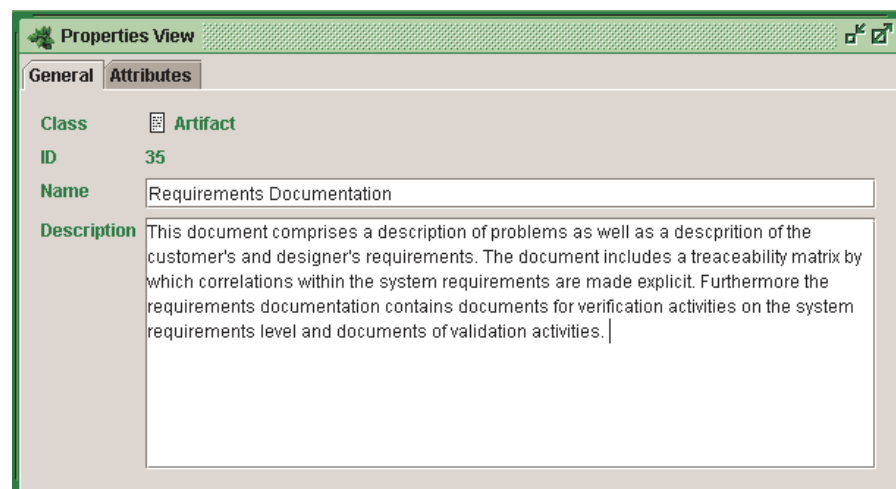


Figure 3.42

Properties view, general panel

- Choose the attributes tab
- Click 'Add' to create a new attribute for this specific artifact
- Click modify and specify name, type and value (

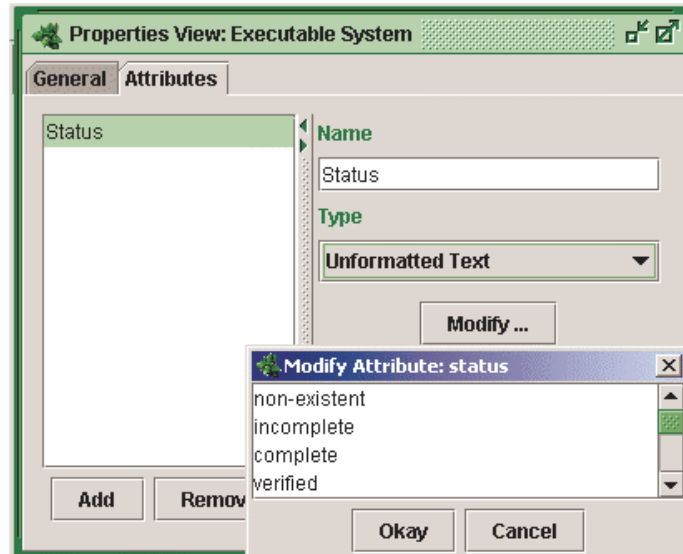


Figure 3.43

Properties view, attributes panel

The properties window can be closed via the main menu bar. To do this

- Go to the Window menu and disable the check box 'Properties View' or use the key Control+D.

Something to remember: Use the properties view for entering additional information about process entities you will later on display in your process handbooks, such as tasks needed to perform an activity, links to examples or templates for artifacts, or required skills for roles. This information appears in the process handbooks and EPGs generated from SPEARMINT™ process models.

3.10 Generating Process Guides

A process guide is a reference document for an intended process, providing guidance to process participants in carrying it out. SPEARMINT™ provides facilities to generate electronic and paper based Process Guides.

- Click on the process model
- Choose the option 'Generate' from the pop up menu
- The following options are available: Electronic Process Guide, printable documentation, other, SPEARMINT™ 5 EPG

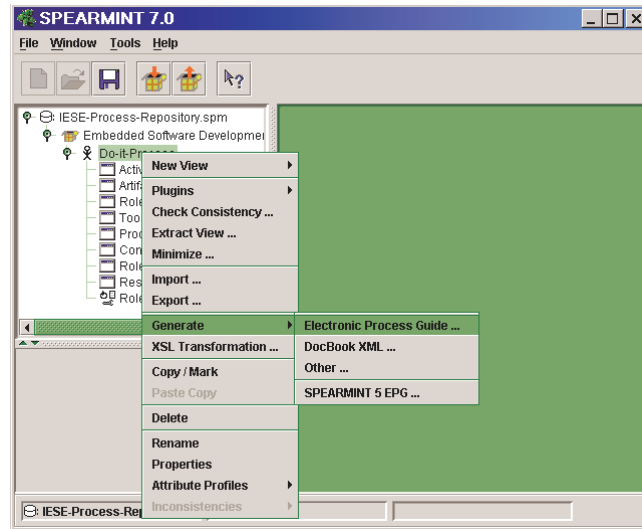


Figure 3.44

'Generate EPG' option on the context menu

3.10.1 Generating an Electronic Process Guide

SPEARMINT™ 's Electronic Process Guide creates an HTML set which helps the user to navigate through the information contained in the guide.

- To generate an EGP, right click on the process model 'Do-it-Process' for the context menu to pop up
- Choose the option 'Electronic Process Guide':
- In the upcoming dialog 'Select Generated Document Directory', choose an existing folder with a single mouse click and hit the 'Select directory' button on the bottom right.

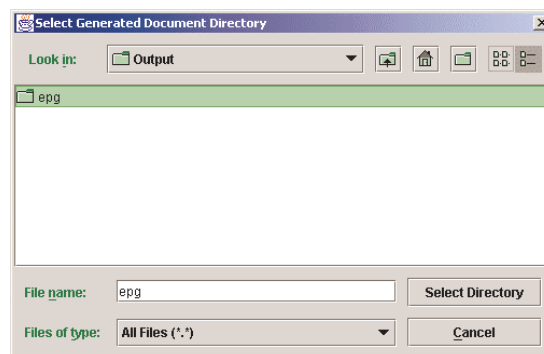


Figure 3.45

The 'Generate EPG' dialog

- The next upcoming dialog 'Deleted Files', shows a warning about the files which should be deleted.

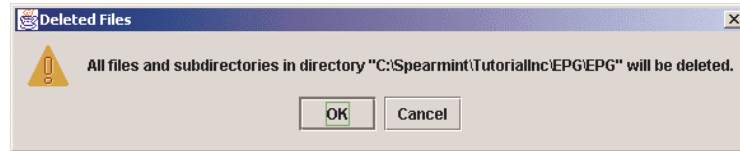


Fig 3.46 Warning dialog

- Accept the warning and SPEARMINT™ will start to generate the EPG.
- SPEARMINT™ has created a new folder for the EPG.
- To load the EPG, open the explorer and go to the EPG file folder.
- Double click the file index.html

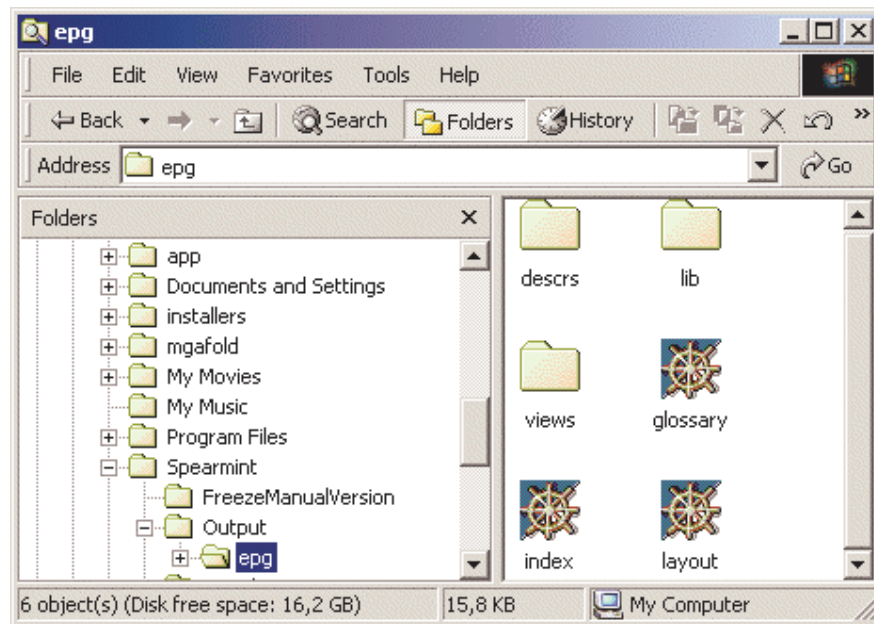


Figure 3.47 Finding the starting point of the EPG: index.html file

The EPG divides the browser window into three frames. The frame on the upper left provides a structural tree with links to each SPEARMINT™ entity. The frame below shows the attributes of the entity selected, and the frame on the right shows the information of the link selected either from the upper or lower frame.

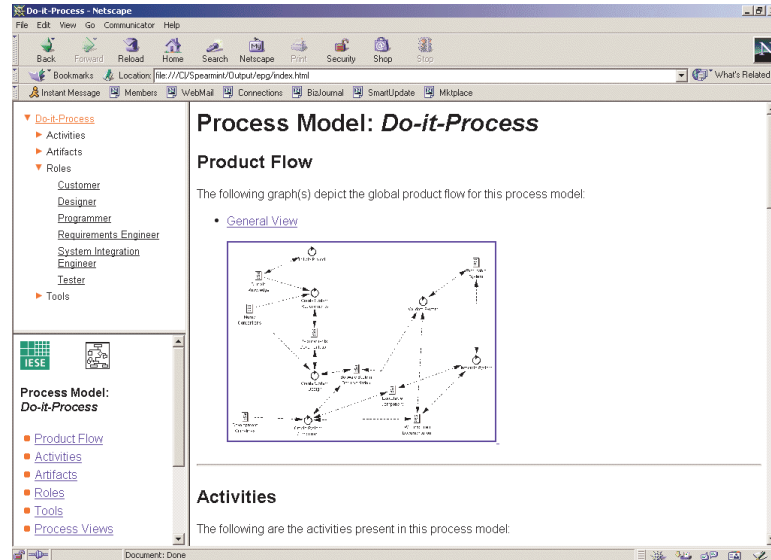


Figure 3.48 The generated EPG

3.10.2 Generating Printable Process Documentation

SPEARMINT™ Printable Process Documentation creates a process guide, which could be printed, and uploaded by some word processors.

- To create a printable process guide right click on the process model 'Do-it-Process' for the context menu to pop up
- Choose the option 'Generate > DocBook XML ...'
- SPEARMINT™ automatically will generate a DocBook XML document
- With this product, the user can generate printed documents. For more information on DocBook please refer to <http://www.oasis-open.org/docbook/>, <http://sourceforge.net/projects/docbook/>, <http://www.oreilly.com/catalog/docbook/>.
- SPEARMINT™ provides a prototypical solution as well: In the main menu, choose the option 'Tools > DocBook XML to PDF ...' and select the generated DocBook XML file. SPEARMINT™ will automatically convert the DocBook XML file to a PDF file.

3.10.3 Generating a Customized Process Guide

SPEARMINT™ provides the user with several templates, which can be used in order to generate documents with different layouts and structure from the default template (Printable process documentation option).

- To generate a Process Guide based on specific templates right click on the process model 'Do-it-Process' for the context menu to pop up
- Choose the option 'Other...'
- Select the file 'Standard-epg.tpl' and click read template

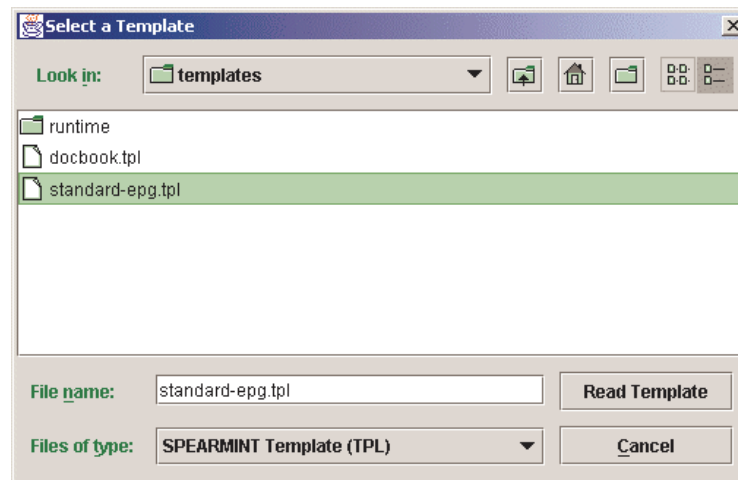


Fig 3.49

Template selection

- Select a generated document directory
- Hit the 'Select directory' button on the bottom right.
- To load the printed process documentation open the explorer and go to the folder you have previously selected
- Open the file index.html generated.

3.10.4 Generating a SPEARMINT™ 5 Electronic Process Guide

- To generate a SPEARMINT™ 5 EPG right click on the process model 'Do-it-Process' for the context menu to pop up
- In the upcoming dialog 'Generate EPG', choose an existing folder with a single mouse click
- Hit the 'Generate EPG' button on the bottom right.

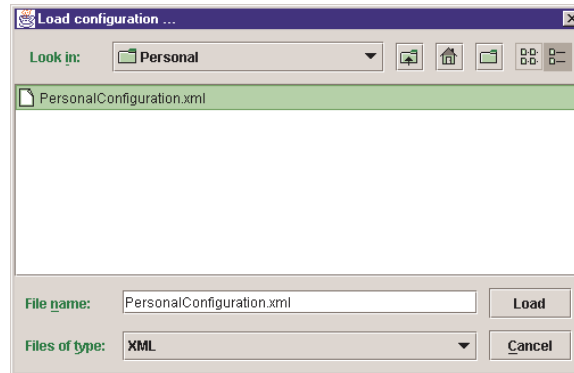


Figure 3.50

The 'Load configuration' dialog

- Choose the configuration file 'PersonalConfiguration.xml'.
- Hit the 'Load' button on the bottom right and SPEARMINT™ will start to generate the EPG (If no file is chosen, simply hit the 'Cancel' button and SPEARMINT™ will choose the default configuration file.)
- On completing the generation, SPEARMINT™ will notify the user where the starting point of the EPG lies.
- To load the EPG, open the explorer and go to the EPG file folder. Double click the file index.html

3.10.5 Transforming an XML File with XSLT

This option can be applied at the database level, project level or process model level. To transform an XML file with XSLT:

- Select the XSLT Style sheet file (.xsl) and click 'Open'
- Select the source XML file and (.xml) and click 'Import'
- Select the target file and click 'Export'
- Open the target file.

This option changes the original XML document based on a fixed DTD, into another customized XML document. This option is for users with knowledge of SPEARMINT™'s DTD and XSLT [W3C].

4 User Interface

This section gives a detailed description of the elements contained in SPEARMINT™'s graphical user interface.

4.1 General User Interface Structure

The picture below shows the user interface of SPEARMINT™. On top of the window are a *menu bar* and a *toolbar*. On the bottom is the *status bar*.

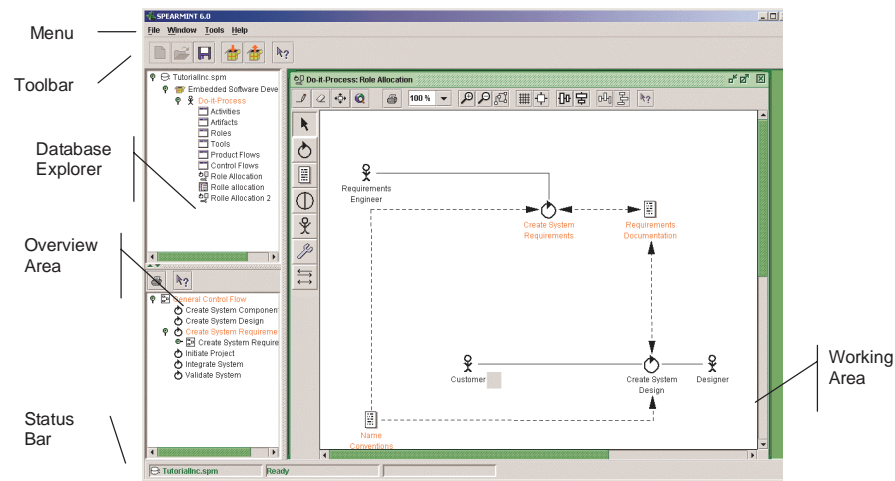


Figure 4.4-1

Screenshot SPEARMINT User Interface

The middle section of the user interface contains the *database explorer*, the *overview view area* and the *working area*.

In the database explorer on the upper left side, the user can navigate through all views of a process and the different projects stored in a database. The database explorer has a treelike structure with the database as root element.

By double-clicking an element in the treelike structure of the database explorer the overview area displays its contents.

By right clicking on the element the contents are shown as a window in the working area.

The working area shows all the views that have been opened, but it's only possible to work with one at a time.

4.2 View Manager

In the database explorer, all project models and views are shown as a tree. The common root is the opened database. The process models contain eight predefined views (showing activities, artifacts, roles, tools, product flows, control flows, role involvement, and resource usage).

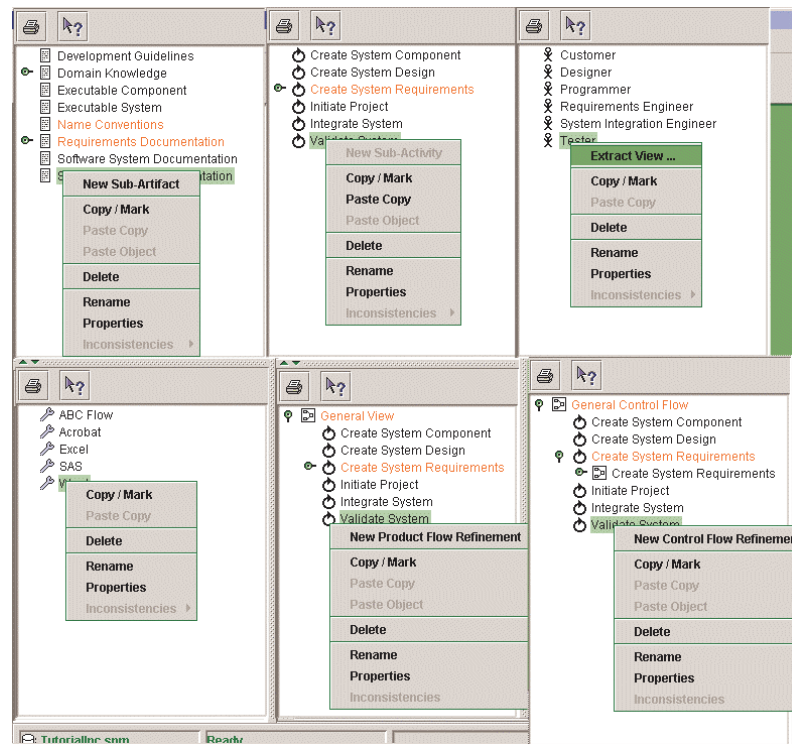


Fig 4.2

SPEARMINT™'s views

Additionally, user-defined graphical views and hypertext views are registered here. Each type of node in the tree (database, project, process model, and views) has its own popup menu, which can be opened by using the right mouse button.

4.3 Working Area

The working area contains all views that are opened. These views can be moved, maximized, minimized, and closed.

The views also have their own popup menus.

The views can be fixed outside the working area in order to maintain the window visible. This is done by double clicking the view to be fixed in the database explorer.

4.3.1 Menu, Toolbar and Status Bar



Fig 4.3

Menu bar

4.3.1.1 Menu bar

The menu bar contains the following menus:

- **File Menu:**

The menu items for opening an existing database or creating a new database are only active if no database is open.

The menu items for closing or saving a database, as well as for importing and exporting, are only active if a database is open.

To export a database

- Choose the option 'Export..'
- Select the directory where the project's information is going to be exported
- Name the file
- Click the 'Export' Button

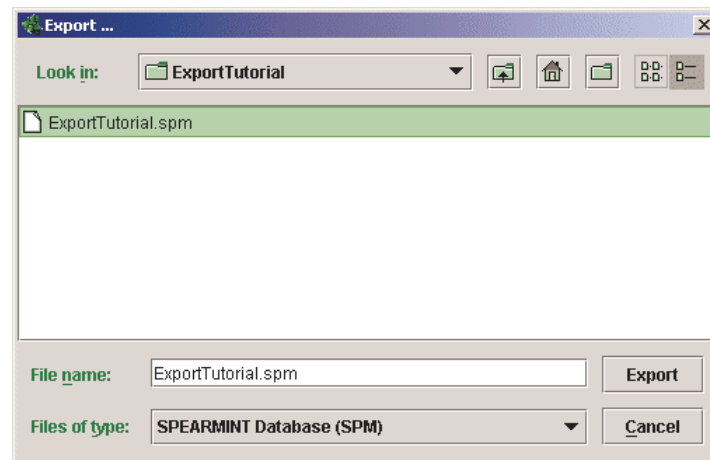


Fig 4.4

Exporting a database

To import a database:

- Choose the option 'Import...'
- Select the directory where the project's information is going to be taken
- Select the file

- Click the 'Import' Button

There is a choice of accessible databases that can be opened when no database is open and the option to exit the program.

- **Window Menu:**

Contains a feature for arranging the windows in the working area and a toggle for showing or hiding the properties window.

All opened windows in the working area are listed, and by clicking the toggle of a specific window, that window is raised.

- **Tools Menu:**

XSL Transformation changes an XML document based on a fixed DTD, into another customized XML document through the use of a XSLT template. This option is for users with knowledge of SPEARMINT™'s DTD and XSLT [W3C].

DocBook XML to PDF transforms a DocBook XML file into a PDF file.

The option **Consistency Assurance** opens a dialog where the user can adjust the behavior of SPEARMINT™'s consistency management.

This window contains very specific information and should be reviewed carefully. The details are written in the chapter SPEARMINT™ 'Consistency Rules'.

The menu item **Options** opens a dialog to customize the look and feel of the user interface and your HTTP proxy settings. Correct HTTP proxy settings are required for online problem reporting to work properly.

To change the look and feel of SPEARMINT™'s interface

- Choose the option 'Options'
- Select the look and feel and the preferred language (English or German)
- Click Ok

To change the HTTP proxy settings

- Choose the option 'Options'
- Click on 'Proxy' and enter the proxy host and port information.
- Click Ok

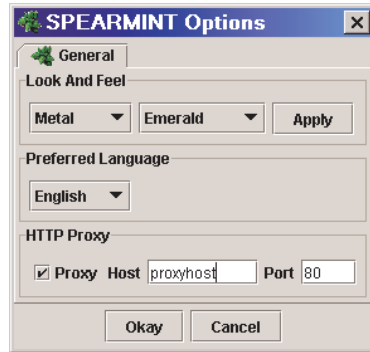


Fig 4.5

SPEARMINT™ 's look and feel.

- **Help Menu:**

The help menu opens the online documentation and provides a form to report bugs. The 'About SPEARMINT™ ' information can be obtained here also.

To report bugs in SPEARMINT™

- Choose the option 'Report Problem'
- Fill the form and click 'Submit'

SPEARMINT/EPG

Software Process Elicitation, Analysis, Review and Measurement in an **I**ntegrated Modeling Environment

SPEARMINT is a registered trademark by Fraunhofer Institute Experimental Software Engineering (IESE).

REPORT A PROBLEM OR REQUEST A CHANGE

Please see the [field descriptions](#) if you are not sure how to fill out this form

Organization: unknown

Originator: (email address)

Class: sw-bug

Severity: non-critical

Priority: low

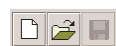
Release: SPEARMINT 6.0

Synopsis of the problem or issue:

Fig 4.6 Report Problem

4.3.1.2 Toolbar

The toolbar contains the following buttons.



These buttons manipulate the database.

The two left buttons create a new database or open an existing database.

They are only active if no database is open.

To close a database, use the file menu entry 'close'.



The save button on the right is only active if a database is open. Then the new and open database buttons are disabled.



With these buttons, you import process model data into the open database or you export process model data from the open database.



With the Help button you activate the help system (only available in German).

4.3.1.3 Status bar



Fig 4.7

Status bar

The status bar displays information about the currently loaded database, the status, and a progress indicator if SPEARMINT™ is busy.

4.4 Popup Menus

4.4.1 Database Popup Menu

With the popup menu of the opened database, the following options are available:

- **New Project...** creates a new project
- **Plugins...** This option is intended for research purposes as an extension point from SPEARMINT™. The programmes attached to this function are mainly developments which enhance specific aspects around process modelling. Therefore the contents and functionality are beyond the scope of this manual.
- **Import...** one or more projects into your database. For importing files, two formats are available, XML and the proprietary SPEARMINT™ file format SPM
- **Export...** a complete database with many projects as one XML – file. This is especially important when migrating from one SPEARMINT™ version to the next
- **XSL Transformation...** changes the original XML document based on a fixed DTD, into another customized XML document. This option is for users with knowledge of SPEARMINT™'s DTD and XSLT.[W3C]
- **Paste Copy...** inserts a copy of a project from the clipboard
- **Properties...** opens the properties window in the working area

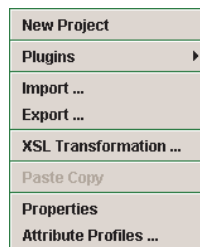


Figure 4.8

Database popup menu

- **Attribute Profiles...** Selects the project profile that had been set up previously.

4.4.2 Project Popup Menu

With the popup menu of the project, the following options are available:



Figure 4.9

Project popup menu

- **New Process Model** creates a new process model
- **Similarity Analysis...** allows for comparisons between two different process models.
- **Import...** one or more process models into your database. For importing files, two formats are available, XML and the proprietary SPEARMINT™ file format SPM
- **Export...** a complete process model as one XML – file
- **Copy/Mark...** makes a copy of the project
- **Paste Copy...** inserts a copy of a process model from the clipboard
- **Paste Object...** moves the process model on the clipboard into this project
- **Delete** and **Rename** the project.

The other options are similar to the database popup menu.

4.4.3 Process Model Popup Menu

With the popup menu of the process model node, the following options are available:

- **New View ...** adds new process and hypertext views
- **Check Consistency....** shows the violation of the consistency rules according with the definition made in the Consistency Manager.
- **Extract View...** creates a copy of this process model in which it removes all entities that are not connected with a specific role. To extract a view:
 - A dialog appears where the user can select the desired role
 - A new process model is created automatically and named as the original connected with ":" to the role selected. Double click the process model.

- Notice that the views in the new process model are a subset of the original. They contain only the activities, artifacts, roles, tools, and split/join states directly related to the selected role.

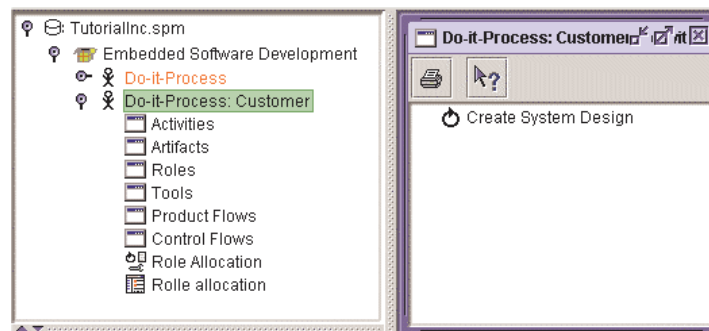


Fig 4.10

Extract view

- **Minimize** deletes all artifacts, roles, tools, product flow graphs, and control flow graphs that are not connected to any activity.
- **Generate** This option allows the user to generate Electronic Process Guides (EPG), using different templates, as inputs, and producing different types of documents as outputs.
 - **Electronic process guide ...:** This option allows to generate an HTML based electronic process guide from the process, using SPEARMINT™'s default template. For a detailed description please see chapter 7.
 - **DocBook XML ...:** This option allows to generate a printable process documentation using an open source standard DTD for books called DocBook. Its main structures correspond to the general notion of what constitutes a "book,". For more information on this template please refer to www.docbook.org. To generate 'Printable process documentation' select the option and SPEARMINT™ will generate automatically the document.
 - **Other ...:** This option allows the use of SPEARMINT™ templates which present the documentation of the process in different ways. The templates are defined using an XML based standard. The user can also generate its own templates. The steps to generate the templates and the reference for the XML notation to use are best described in the "Guidelines to the generation of customizable process guides in SPEARMINT™".
 - **SPEARMINT™ 5 EPG ...:** This option allows to generate an HTML based electronic process guide using SPEARMINT™ 5.0's generator. For a detailed description please see chapter 7.
- **Copy/Mark...** makes a copy of the process model

- **Paste Copy...** inserts the process model on the clipboard into this process model.
- **Attribute Profiles...** activates the attributes which have been defined previously for the project

4.5 Popup Menus of the Predefined Views

Each predefined view (see section 2.4) in the process model has a general background popup menu as well as a specific node popup menu of the entity presented in a specific view.

The popup menus are almost alike and only differ in one or two menu items.

4.5.1 Background Popup Menu

All hierarchical views have a background popup menu that is the same except for the first menu item.

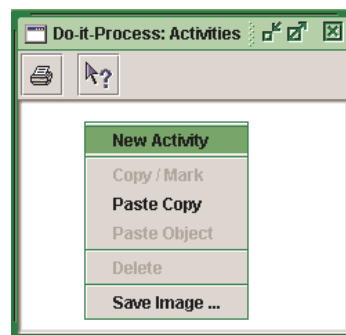


Figure 4.11

Background popup menu of the activity view

- **Copy / Mark, Paste Copy, Paste Object** and **Delete** are the standard functions for process elements.
- **Save Image** stores an image of the view as a *.jpeg, *.gif or *.cgm file for documentation purposes.
- The option **Paste Copy** inserts a copy of entities into a view. The feature can also be used across different process models.

- The option **Paste Object** moves the entity on the clipboard to the specific location in the tree. It is not present in the roles and tools views.

Depending on the view the user can add a new activity, role, artifact or tool using the first menu item (eg. New Activity, New Artifact).

The product flow graph and the control flow graph are allowed to add activities to the model.

The product flow view and control flow view have a menu item for adding a new product flow refinement graph or a new control flow refinement graph

4.5.2 Node Popup Menu

The popup menus for the nodes in the activities and artifacts view are the same except for the first menu item.

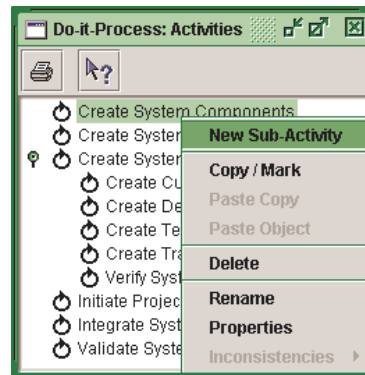


Figure 4.12

Node popup menu of the activities view

- The first menu item **New Sub-Activity** (depends on the view) adds a new activity or artifact to the currently selected node.
- **Copy / Mark, Paste Copy, Paste Object**, and **Rename** are the standard functions for process elements.
- **Delete** removes an entity from the entire process model. As a result the entity and all its relationships are deleted from the model. The function is irreversible!
- **Properties** opens the properties dialog to edit the attributes and the description of the selected entity.

- **Inconsistencies** shows all inconsistencies related to the selected entity.

The first menu item (**New Sub-**) is only available in the activities and artifacts view.

The roles and tools view do not need these items because they contain only a list of items and no hierarchy.

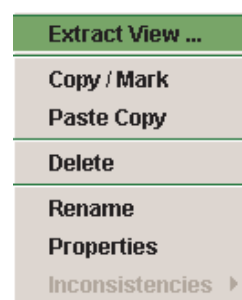


Figure 4.13

Node popup menu of the roles view

The roles view has the additional option **Extract View...**, which extracts the view of a given role with all its links into a newly created process model within the same project.

Again, the roles view and the tools view do not have the option Paste Object in their node popup menu.

4.6 Graphical Views

This section describes the user interface of the graphical views, i.e., user-defined graphical view, product flow graphs, and control flow graphs.

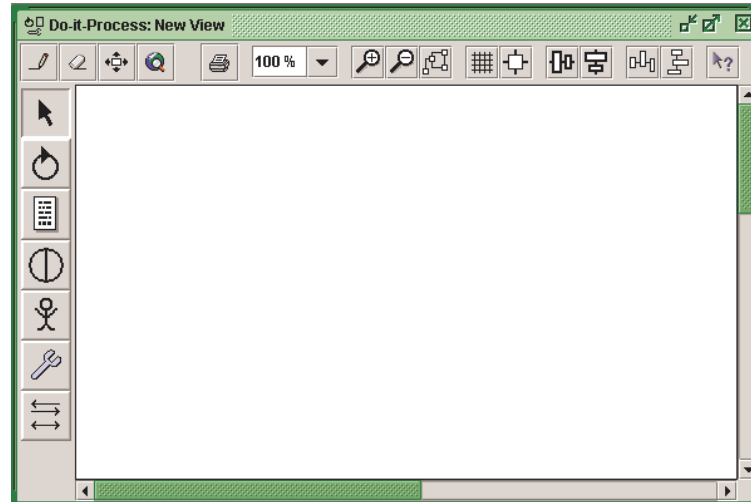


Figure 4.14 User-defined graphical view

The structure of the user interface is the same for all graphical views.

The different tools are located in the vertical toolbar on the left side.

The horizontal toolbar on the top contains some buttons specific to the view and some general buttons for the graphical manipulation of the graph.

The vertical toolbar on the left side and the specific buttons in the horizontal toolbar, are for the description of the process, the product flow graph, or the control flow graph.

4.6.1 General Functions of Toolbars in Graphical Views

The horizontal toolbar contains buttons for including and excluding selected entities. Entities can be selected by mouse click in any view:



Include selected, includes all selected objects into the user-defined graphical view



Exclude selected, excludes all selected objects from the user-defined graphical view




Include neighbors, includes all objects that have an association with the selected object



Include all, includes the whole process model

The top button of the vertical toolbar is activated when a graphical view has been opened:

-  In order to work on the object in the graph, the select button has to be activated. Now objects can be edited, moved, etc.

You can also create new entities by using the vertical toolbar on the left side. Activate the appropriate button and every mouse click in the working area will create a new entity:



Adds a new activity to the graph.



Adds a new artifact to the graph.



Adds a new role to the graph.



Adds a new tool to the graph.



Links can be added by holding the left mouse button and moving the mouse to the object that you wish to link it to. To create a modifying link repeat this procedure in both directions.

The horizontal toolbar also contains buttons for zooming, distribution, alignment, toggle grid, and snap nodes to the grid. A node here is any object ,which has been drawn. (Artifact, activity, tool, role, link, join/split status).

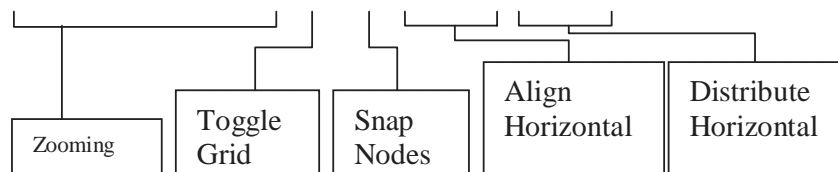


Figure 4.15

Horizontal Toolbar

- **Zooming:** You can zoom in and out with the magnifying glass icon.
With the combo box, you can select a zoom value or type in an individual value.
The right button fits the whole drawing to the visible window size (very useful if you get lost in a graph).
- **Toggle Grid:** Toggles the visible grid on / off. The grid can be customized.
- **Snap Nodes:** Toggles the snapping of the nodes on / off. The style for the snapping can be customized.
- **Align Horizontal / Vertical:** Aligns all selected nodes horizontally or vertically with the default alignment strategy.

- **Distribute Horizontal / Vertical:** Distributes all selected nodes horizontally or vertically with the default distribution strategy.

4.6.2 Background Popup Menu (only for Graphical View, Product Flow Graph and Control Flow Graph)

User-defined graphical views, product flow graphs and control flow graphs have similar popup menus that can be invoked from the background.

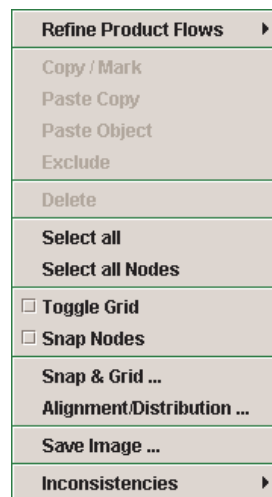


Figure 4.16

Popup menu of the product flow view

- **Refine Product Flows** (only product flow graphs) allows to include consumed and produced artifacts or the sub-artifacts (see 5.2.5 for details).
- **Copy / Mark, Paste Copy, and Paste Object** have the standard function.
- **Exclude:** removes the selected entities from the graph but does not delete them from the model.
- **Delete:** deletes all selected elements from the model. These elements are irrecoverable.
- **Select all:** selects all objects in the view
- **Select all Nodes:** selects all nodes in the view

- **Toggle Grid:** toggles the grid on / off. If the grid is on, the checkbox it is marked.
- **Snap Nodes:** toggles the snapping of the nodes on / off. If the snapping is on, the checkbox it is marked.
- **Snap & Grid ...:** opens the dialog for customizing the snap style, the grid for snapping, and the visible grid (see 4.6.4)
- **Alignment/Distribution ...:** opens the dialog for vertical/horizontal alignment and distribution (see 4.6.5)
- **Save Image ...:** opens a dialog where you can save the graph as a graphic file
- **Inconsistencies:** shows all inconsistencies of elements contained in this view

4.6.3 Edges Popup Menu

With a right mouse click on an edge or a routing point of the edge, a popup menu is displayed.

The Delete Routing Point is active if the menu is invoked on a Routing Point

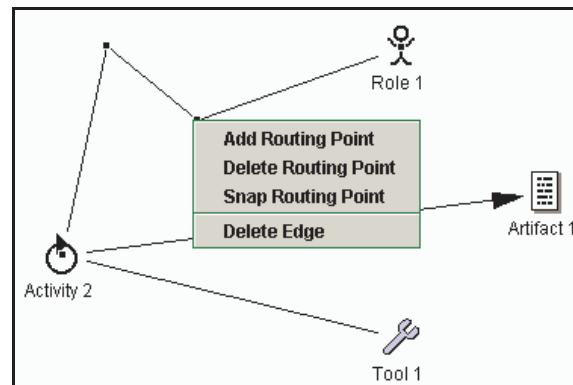


Figure 4.17

Popup menu of the edges

- **Add Routing Point:** adds a routing point at the location where the popup menu was invoked. Selecting it and moving along the link can translate the point.
- **Delete Routing Point:** deletes a routing point and draws a straight line between the right and left neighbors of the old point.
- **Snap Routing Point:** translates the routing point to the nearest orthogonal position of one or both neighbors.

If the menu is invoked on a routing point, both neighbors are used.

If the menu is invoked on the edge, then the nearest point to the invoking position is translated orthogonal to its neighbor.

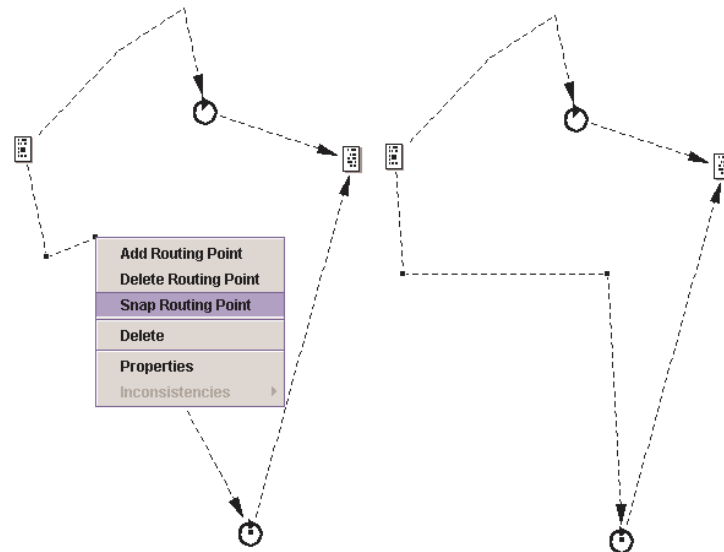


Fig 4.18 Snap routing point to nearest orthogonal

- **Delete Edge:** deletes the edge from the graph.

4.6.4 Grid and Snapping

The graphical view has two grids:

- One that is only visible and has no further effect
- One invisible that manages the snapping of the nodes.

The nodes can snap to the invisible grid with different strategies and with different anchor points. The anchor points are shown below.

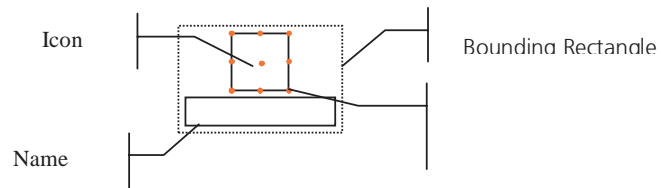


Figure 4.19 Position of the anchor points for snapping

Depending on the strategy, several or all anchor points are inactive.

Only for the Snap to Borders strategy, all anchor points are disabled because the bounding rectangle snaps to the horizontal and vertical gridlines.

If snapping is turned on, all nodes stay in their original position. If new nodes are added, these are placed ignoring the grid. They snap only if the user selects or drags one or several nodes.

Customize the snapping strategy:

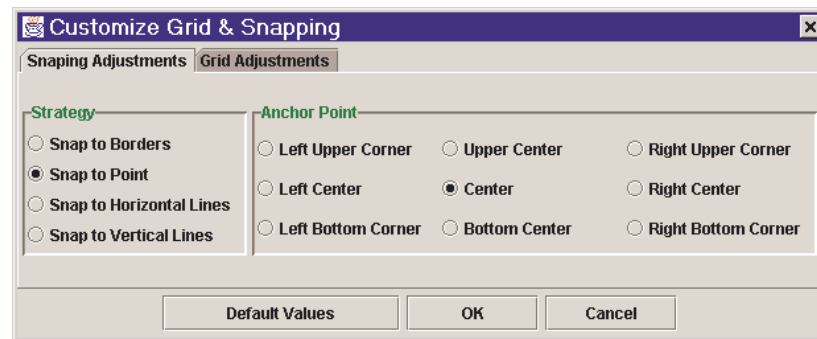


Figure 4.20

The Snapping Adjustments Tab

In the Snapping Adjustments tab (Figure 4.2), the user can customize the strategy of snapping the nodes to the grid.

In the left area the strategy is selected, in the right the anchor point inside the nodes can be chosen (the anchor point is oriented on the icon of the node).

Strategies:

- **Snap to Borders** snaps to the borders of the bounding rectangle (all anchor points are disabled).
- **Snap to Point** snaps to the selected anchor point (all anchor points are enabled).
- **Snap to Horizontal Lines** snap the upper center anchor point, center anchor point or bottom center anchor point to the horizontal lines.
- **Snap to Vertical Lines** snaps the left center anchor point, center anchor point or the right center anchor point to the vertical lines.

Customize the Grid Adjustments:

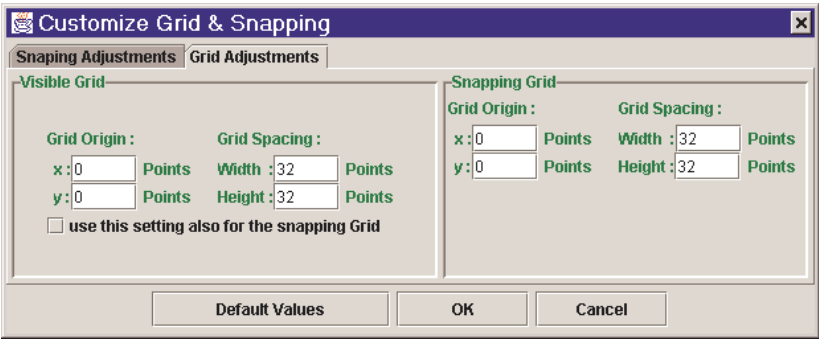


Figure 4.21

The Grid Adjustment Tab

In the Grid Adjustment tab (Figure 4.2), the user can customize the visible grid and the snapping (invisible) grid.

In the left area are all properties of the visible grid and also a checkbox for using this setting for the snapping grid as well. When this option is selected, the input fields for the snapping grid in the right area are disabled.

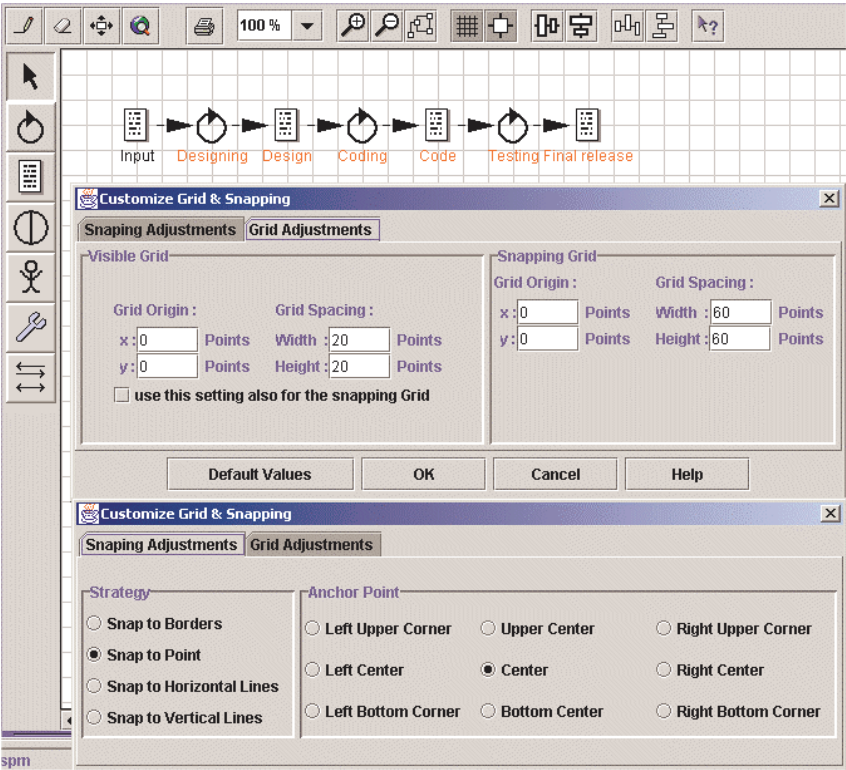


Fig 4.22

Snap and grid strategies' example

The previous picture shows an example of a strategy. The nodes in the view are snapped to the center point of the snapping grid every 60 points. The difference between nodes can be seen as tree squares in the visible grid.

4.6.5 Align and Distribute

This dialog combines two different actions, each for vertical and horizontal orientation. Thus, the dialog contains four tabs, align horizontal, align vertical, distribute horizontal, and distribute vertical. All these actions affect only the currently selected nodes if the OK button is pressed.

The alignment and distribution is effected if the snapping to the grid is on. If the snapping is on, the nodes snap to the nearest point that is computed for the translation.

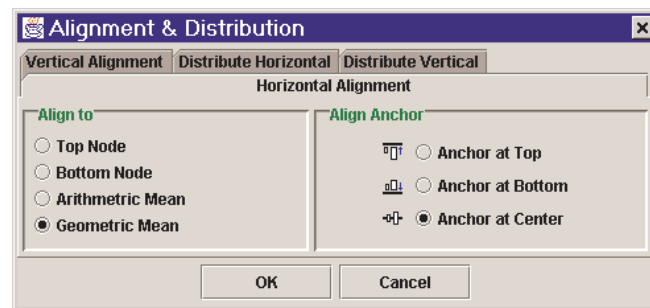


Figure 4.23

Horizontal Alignment Tab

In the Horizontal Alignment tab (Figure 4.2), the user can select the strategy of how the nodes should be aligned.

In the left area, the user selects the strategy for the alignment:

- **Top Node:** translate all other nodes to build a horizontal line with the topmost node
- **Bottom Node:** translate all other nodes to build a horizontal line with the bottommost node
- **Arithmetic Mean:** translate all nodes to build a horizontal line at the arithmetic mean of all selected nodes
- **Geometric Mean:** translate all nodes to build a horizontal line at the geometric mean of all selected nodes

In the right area, the user can select whether the center, the top or bottom edge of the icon is the reference point for the alignment. This is necessary if the icons have different sizes.

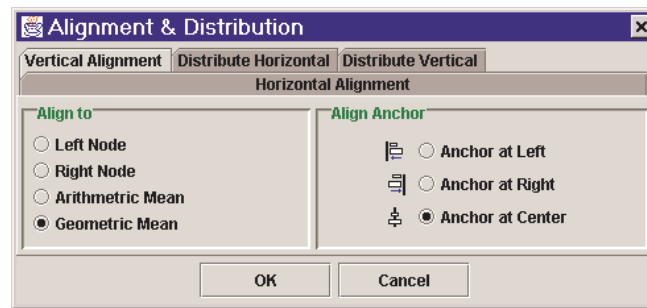


Figure 4.24

Vertical Alignment Tab

The Vertical Alignment tab (Figure 4.2) works like the Horizontal Alignment tab.

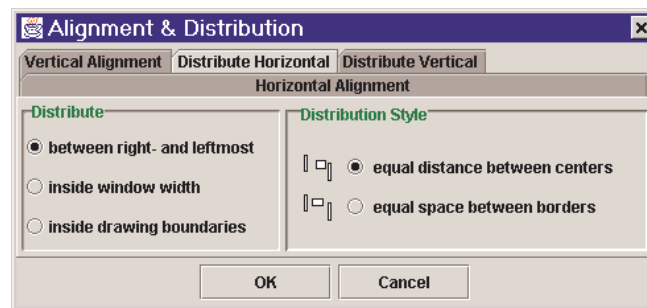


Figure 4.25

Distribute Horizontal Tab

In the Distribute Horizontal tab (Figure 4.2), the user can select the strategy of how the selected nodes should be horizontally distributed.

In the left area, the user selects the range that should be used for the distribution.

- **Between right- and leftmost:** uses the rightmost and leftmost nodes as the border, those are not translated
- **Inside window width:** uses the right and left border of the visible window
- **Inside drawing boundaries:** uses the whole scrollable area of the view

In the distribution style, the user can decide if the nodes always have the same distance between their bounding rectangles or between the centers of the icons. This is relevant if the icons of the nodes have different sizes.

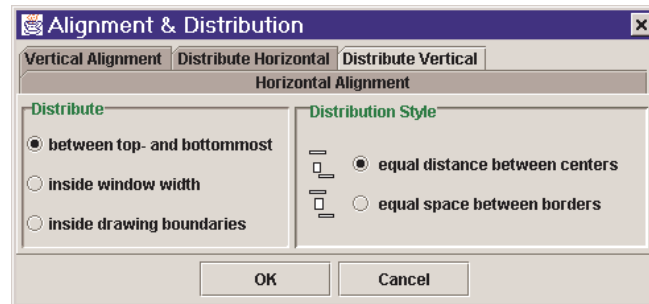


Figure 4.26

Distribute Vertical Tab

The distribute vertical tab (Figure 4.27) works like the distribute horizontal tab.

5 Modelling with Views


5.1 Process Model Edition

This section describes some manipulation functions that work on the entire process model, across views. The ways these functions work are implications of the view concept.

5.1.1 Selection

Any object can be selected by mouse click.

To select more than one object, use the Control key in addition to the mouse click.

In graphical views the select button () has to be activated in order to select objects. Additionally, a select frame can be drawn in graphical views by holding the left mouse button and dragging it across the objects that are to be selected.

Selection acts on the underlying process model and is not based on the view. When you select something in a view, it is selected in the model and its selection is propagated to all views where the entity is visible. This means you can make selections in many views.

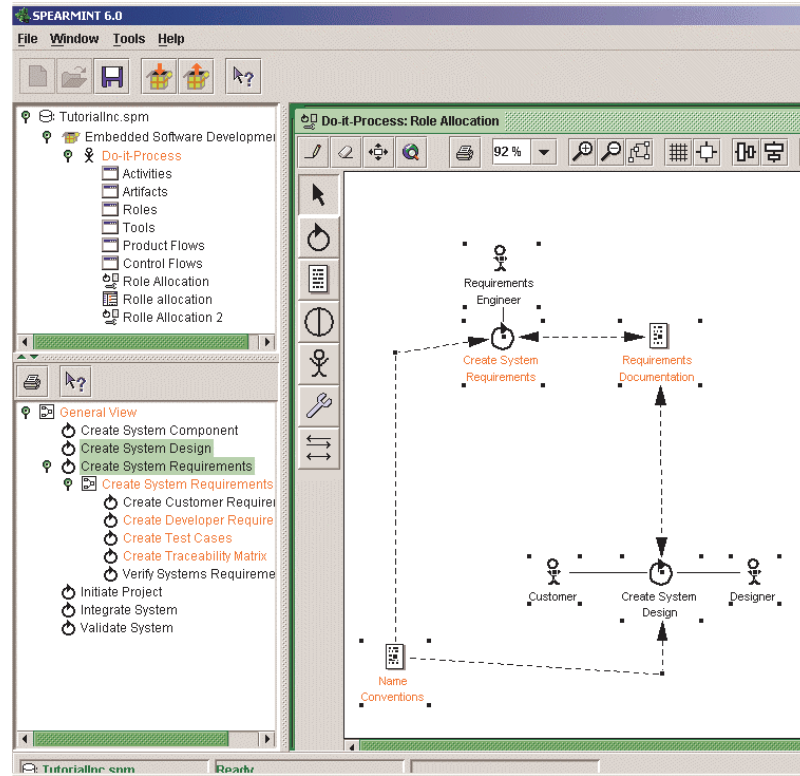


Fig 5.1

Selection

The selection determines what is shown in the properties view. If only one entity is selected, then its properties will be shown in the properties view.

- Selection acts on the process model, not on the view.
- When you select something, it is selected in every view in which it is included and the properties view.

5.1.2 Include and Exclude

For user-defined graphical views and hypertext views, the user can specify what is shown in the views. This is done indicating which elements existing in the process model should be included in the views.

Elements can also be excluded from the views when they are no longer needed.

These operations only act to determine the content of the view and have no effect on the underlying process model.

5.1.3 Delete

- **Deleting** an entity removes it from the entire process model. As a result, the entity and all its relationships are removed from all views where it was showing. The entity is irrecoverable!

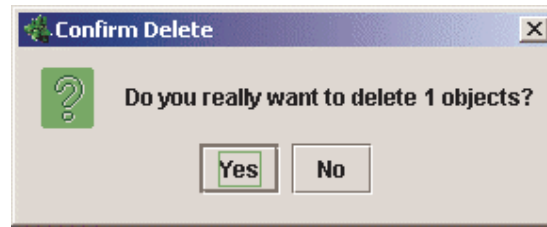


Fig 5.2

Deleting

- **Excluding** removes an entity from its view.

Something to remember:

If an activity is excluded from a refinement graph, however, it will be moved to the top-level of the hierarchical structure in the product and control flow view as well as in the activities view.

It is important not to confuse delete with simply excluding an element from showing in a view.

The delete operation will delete the element from the underlying process model and as a result, it will no longer show in any view in which it was visible.

The exclude operation just removes the element from a particular view, while it continues to exist in the process model.

5.1.4 Copy / Mark, Paste Copy, and Paste Object

The menu item **Copy / Mark** does two things, it copies the selected item(s) of the process to the clipboard, and it also marks it/them.

Now that the object(s) are buffered, you can paste the element(s) in two ways, with **Paste Object** or **Paste Copy**.

If you initiate a **Paste Object** action, the element is included in the view where the command is executed.

For example, if you mark some entities in a product flow refinement view and want to import them to a user-defined graphical view, you have to use Paste Object.

If you initiate a **Paste Copy** action, the underlying model will be changed and changes may be seen in other views as a result of the creation of a new entity.

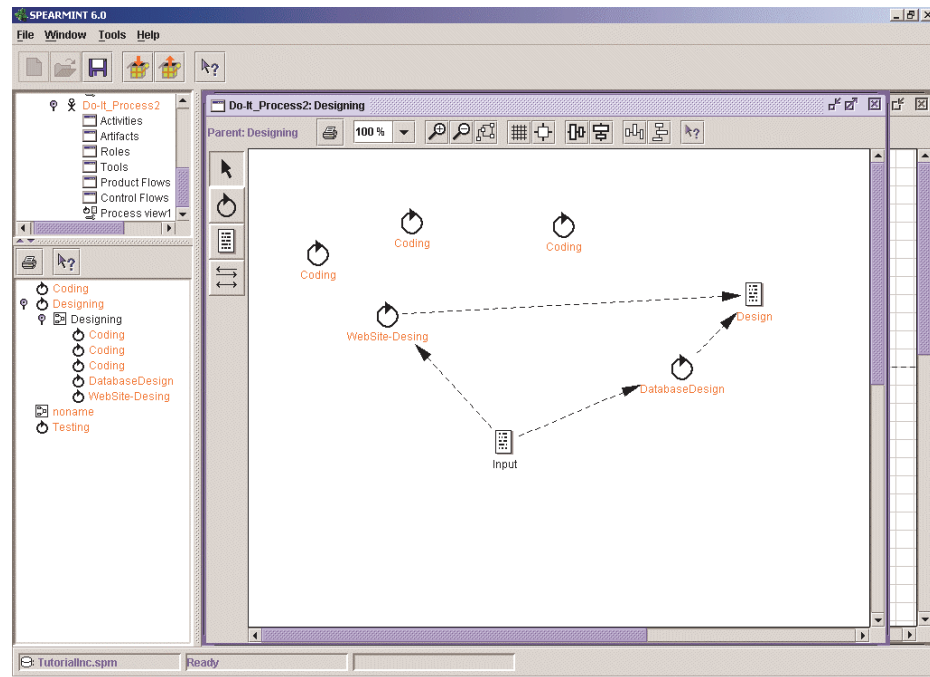


Fig 5.3

Paste/ Copy

When you copy more than one entity, all relationships between these entities are also copied (even if they are not visible in the view). Subsequently, all this information is pasted into the model on a paste action.

- **Paste Object** makes a new (independent) copy of the entity, thus a new entity is created.
- Do not confuse this with “include”, which is the same entity made visible. **Paste Copy** works on the model and not on the view.
- When you copy an entity, you get not only the entity but all of its “context” as well (including attributes and relationships it has to other entities).

It is important not to get confused because the **Paste Object** does only add references to additional views, whereas **Paste Copy** creates new instances of process elements and adds them to the process model.

5.1.5 Tree Structure Modification

For moving entities, use **Copy / Mark** to write the entity in the clipboard. With **Paste Object** the entity is moved to the new location.

When you move elements around in a hierarchy (either the activity or artifact containment hierarchies or the activity refinement structure), the previous

relationship the element had with its parent element is deleted. This is to ensure that the consistency rules are not violated.

When you move something, all of the structure beneath that element moves with it.

- The relationship with the parent is deleted.
- The hierarchy under the entity moves with it.

5.2 Views

5.2.1 Activity View

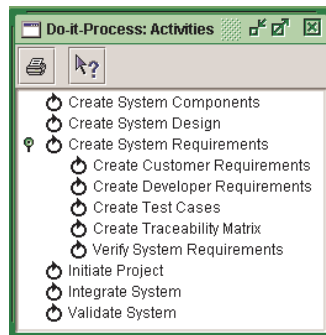


Figure 5.4 The Activities View

The activity view displays activities only in a hierarchical list. It shows all activities that are defined in the process model.

They are displayed in alphabetical order and the sub-activities are displayed as sub-nodes of the tree. The tree can be expanded and collapsed.

All activities that are created in this view will automatically be shown in the hierarchical structure of the product flows view and the control flows view.

5.2.2 Artifact View

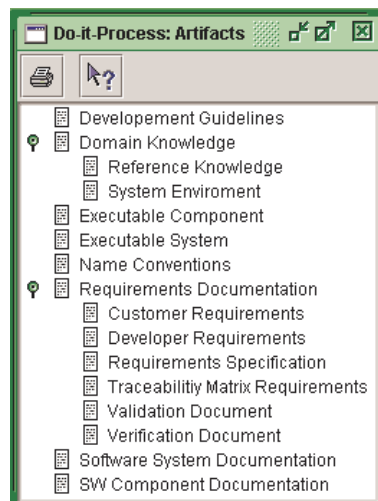


Figure 5.5 The Artifacts View

The artifact view displays artifacts only in a hierarchical list. It shows all artifacts that are defined in the process model.

They are displayed in alphabetical order and the sub-artifacts are displayed as sub-nodes of the tree. The tree can be expanded and collapsed.

For the functionality of the menus, see section 4.5.

5.2.3 Roles and Tools View

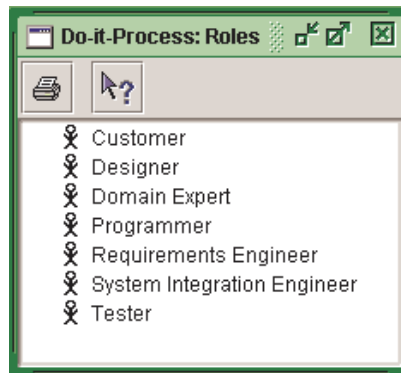


Figure 5.6

The Roles View

The role view and the tool view are simple non-hierarchical list views.

They contain all roles and tools, respectively, that are defined in the process model.

The only other view that can include roles or tools is the user-defined graphical view.

The role view allows the option **Extract view...** of a selected role to get the role-specific view on the process as a new independent process model.

The extracted view will show all entities and their relationships with respect to one specific role. See section 4.5.

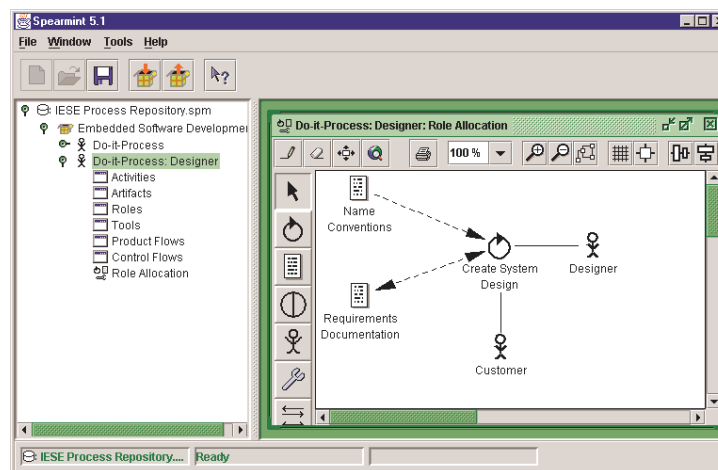


Figure 5.7

Extracted view

5.2.4 Product Flows View

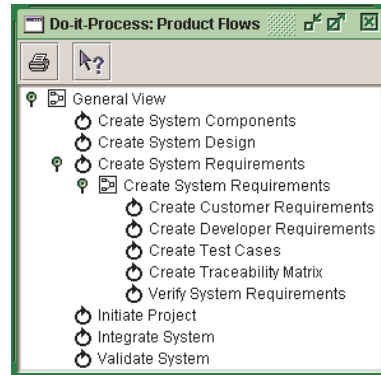


Figure 5.8

The Product Flow Refinement View

Sub activities are included in the product flow refinement graph of their container activity. The graph will automatically be named after the container activity.

To move any activity within the hierarchical structure of the tree may result in a change of the corresponding product flow graphs, i.e., activities moved will be excluded from the graph along with their links to other objects.

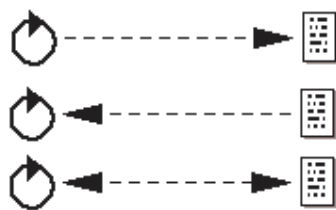
All activities and product flow graphs are contained in the hierarchical structure of the product flow refinement view.

It shows activities and product flow graphs and the refinement relationships between them, as well as which activities are contained in which product flow graphs.

New activities added to the product flow refinement view will automatically be shown in the activity view showing the same hierarchical structure of the activities as the product flow view.

5.2.5 Product Flow Graphs

The product flow graphs contain activities, artifacts, and the relationships between them:



The activity produces the artifact.

The activity consumes the artifact.

The activity modifies the artifact.

It is not possible to link two activities or to link two artifacts directly.

If the product flow refines an activity, the user can include the consumed and produced artifacts of the parent activity.

The activities in a product flow graph can be decomposed to get a finer grained description of the activity.

To create a decomposition of an activity, you can use the popup menu **New Product Flow Refinement** of the activity. This creates a new product flow graph where artifacts and activities can be easily included via the background popup menu.

If a refinement exists for the selected activity, the **New Product Flow Refinement** menu item is deactivated, and the menu items **Show** and **New Activity** are activated instead.

Show opens the diagram for this activity and New Activity creates a new sub activity of the selected activity in the model.

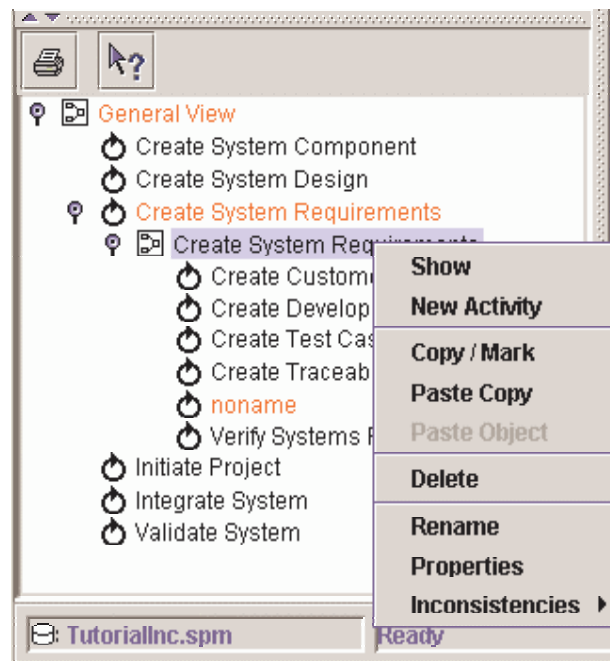


Fig 5.9

Show/New activity options

On the left upper corner of the toolbar is a notice on which activity is the parent activity of that product flow graph. If this view contains a top-level graph, the notice "No parent" will indicate this.

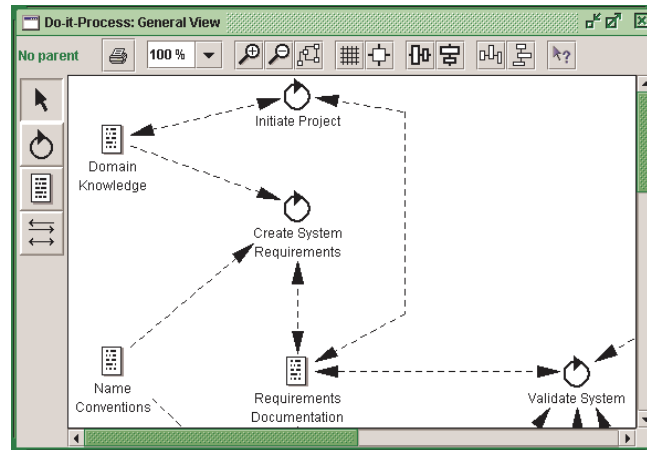


Figure 5.10

A product flow graph

5.2.6 Control Flow View

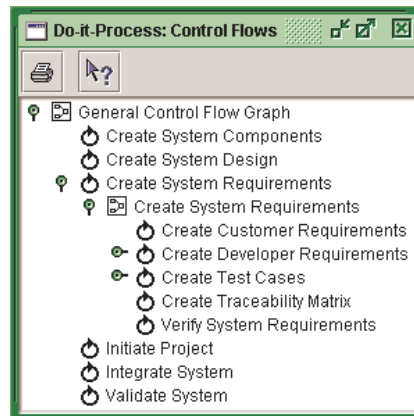


Figure 5.11

The Control Flow Refinement View

All activities and control flow graphs are contained in the hierarchical structure of the control flow refinement view.

It shows activities and control flow graphs and the refinement relationships between them, as well as which activities are contained in which control flow graphs.

New activities added to the control flow refinement view will automatically be shown in the activity view showing the same hierarchical structure of the activities as the control flow view.

The hierarchical structure resembles the containment structure of the activities

and the product flow refinement view.

5.2.7 Control Flow Graphs

The control flow graphs contain activities, join/split states, and precedes links. A join/split state is a circle that is divided in two halves; the left half contains the

join symbol and the right the split symbol. The join and split symbols are AND (x), OR (+), and XOR(\oplus) or nil (empty).

The split symbol is used when an activity is followed by two or more activities:

The join symbol is used when two or more activities precede one activity:

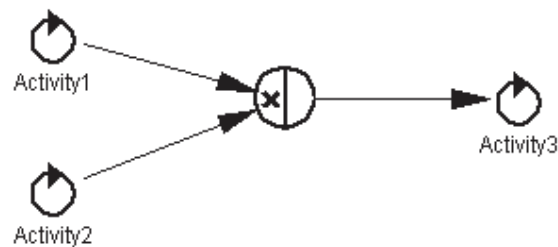


Fig 5.12

Join/And

The join/split state and precedes link were introduced for modeling the control flow graphs. The precedes links are arrows between activities or join/split states.

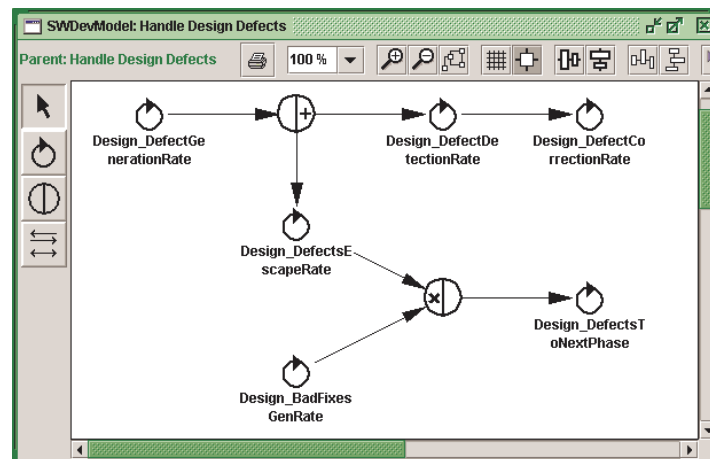


Fig 5.13

Control flow example

Fig 5.13 shows the control flow graph called Handle Design Defects. The description of this graph is as follows:

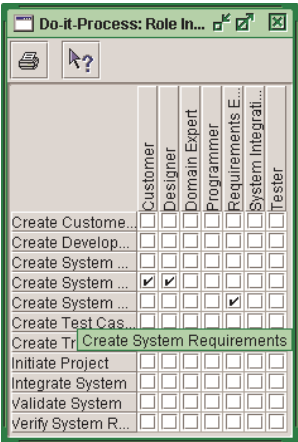
The activity DesignDefectGenerationRate could be followed (OR), by the activity DesignDefectDetectionRate or the activity DesignDefectsEscapeRate. If the branch, which has another join/split state, is followed, then the activity DesignDefectsToNextPhase can be accomplished if and only if, the activities DesignDefectsEscapeRate (AND) DesignBadFixesGenerate have been done before.

5.2.8 Role Involvement View

The role involvement view displays roles and activities in a tabular form. It shows all roles and activities that are defined in the process model.

They are displayed in alphabetical order, without making a distinction between aggregate activities and sub-activities. Names appear abbreviated. Full names are displayed in tool tips only.

The existence of an involvement relationship between a role and an activity is indicated by a check mark in the corresponding table cell.



The screenshot shows a window titled "Do-it-Process: Role Involvement". It contains a table with roles as columns and activities as rows. The roles are Customer, Designer, Domain Expert, Programmer, Requirements Engineer, and System Integrator. The activities are Create Customer..., Create Develop..., Create System ..., Create System ..., Create System ..., Create Test Cas..., Create Tr..., Initiate Project, Integrate System, Validate System, and Verify System R... The table shows involvement relationships with checkmarks in the cells for (Create System ..., Designer), (Create System ..., Domain Expert), (Create System ..., Requirements Engineer), and (Create System ..., System Integrator).

	Customer	Designer	Domain Expert	Programmer	Requirements Engineer	System Integrator
Create Customer...						
Create Develop...						
Create System ...						
Create System ...		✓	✓			
Create System ...				✓		
Create Test Cas...						
Create Tr...						
Initiate Project						
Integrate System						
Validate System						
Verify System R...						

Fig 5.14 Role involvement example

5.2.9 Resource Usage View

The resource usage view displays resources, i.e., tools, and activities in a tabular form. It shows all tools and activities that are defined in the process model.

They are displayed in alphabetical order, without making a distinction between aggregate activities and sub-activities. Names appear abbreviated. Full names are displayed in tool tips only.

The existence of a usage relationship between a tool and an activity is indicated by a check mark in the corresponding table cell.

5.2.10 Graphical View (User-Defined View)

The user-defined graphical view is a special type of view. This is the only graphical view where the user can work with roles and tools.

This view is not part of the refinement hierarchy and can contain parts from each level of the refinement hierarchy. The user-defined graphical view is a partial view.

A user-defined graphical view can contain all four SPEARMINT™ entities, namely activities, artifacts, roles, and tools. Roles and tools can be linked to activities. Activities and artifacts can be linked displaying the relationship of consumption, usage, and modification as described in section 5.2.5. It is not possible to link two activities or to link two artifacts directly.

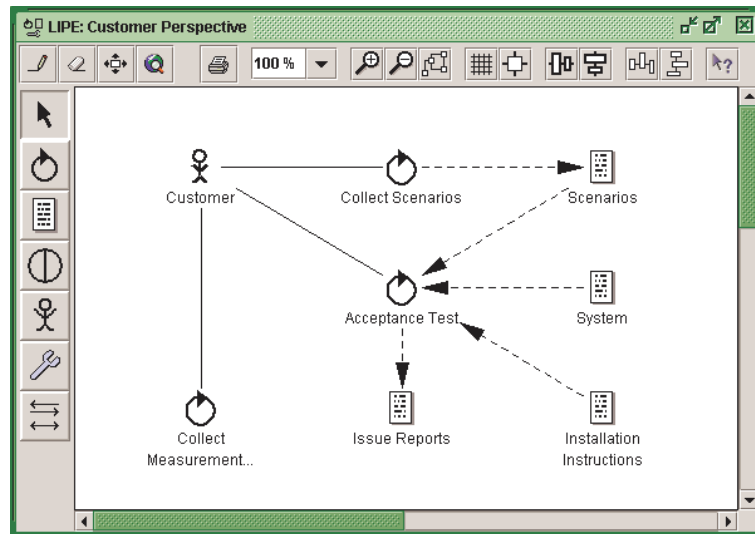


Figure 5.15 A user-defined graphical view

The buttons for including are one way to populate the view with existing activities, artifacts, roles, and tools.

You can also use the popup menu in the background to include entities with **Paste Object**.

With **Paste Copy** the content of the clipboard is included as a copy in the view and the model. That means you created a new entity with the same name and attributes in the model and the view.

The newly created entity appears in the hierarchical view at the root level according to the type of the entity.

Activities also appear in the refinement view; there they can be moved to the appropriate position in the refinement structure.

5.2.11 Hypertext View

The purpose of the hypertext view is to show and discuss a process model with a remote partner. This partner can view the model in a browser (ie. Netscape or Internet Explorer).

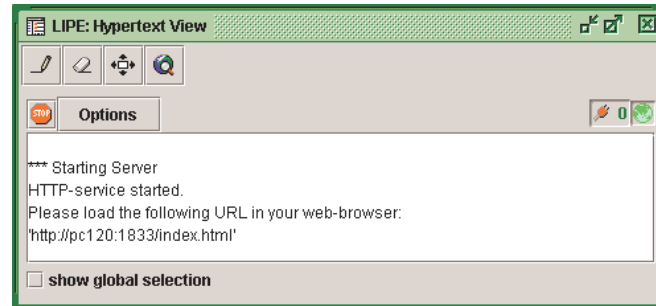






Figure 5.16

The Hypertext view

The user can decide which part of the process model the partner sees. This can be done using the four buttons on the toolbar of the *hypertext view*, which include or exclude specific entities of the process model.

-  Include selected, includes all selected objects to the hypertext view
-  Exclude selected, excludes all selected objects from the hypertext view
-  Include neighbors, includes all objects that have an association with the selected object
-  Include all, includes the whole process model

The button with the stop sign stops the server, and is replaced by a button to start the server again.

When the server is stopped the partner is disconnected. The server must be started again in order to present the process model to the partner.

Every time the server is started in a different location, therefore the partner must be warned about the new location.

On the right side there is a red plug with a number, which indicates how many browsers are connected to the server.

The options button switches the status window of the server to list, with six display options. The user can select which details the participating person could see in the browser.

The following display options can be selected:

- Show description
- Show attributes
- Show product flow

- Show decomposition
- Relations from/to resources
- Relations from/to roles

By clicking on the microphone symbol in the upper frame, the partner can submit a message from the browser to SPEARMINT™. This message appears in the status area of the Hypertext View.

This is an example of a generated view in a browser:

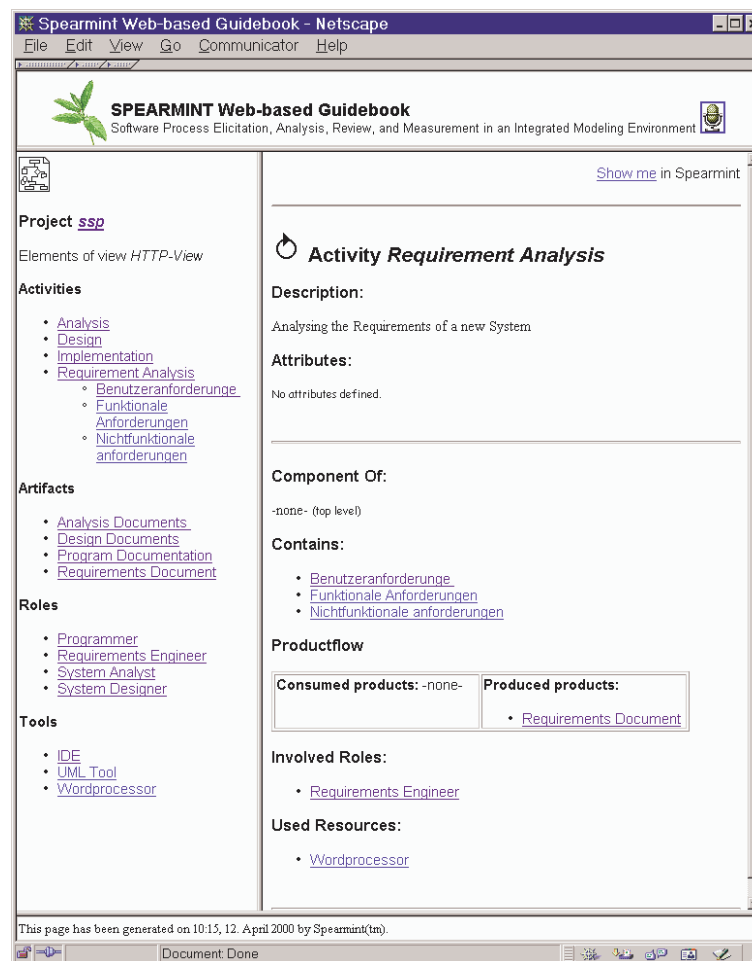


Figure 5.17 The generated HTTP View in the browser.

5.2.12 Properties View

The properties view gives the user the opportunity to type in additional information about the entities. It always displays the information of the entity that is selected. If no entity is selected, the view is empty.

The view has two panels, one for the general properties (see Figure 5.17) and one for attributes (see Figure 5.18).

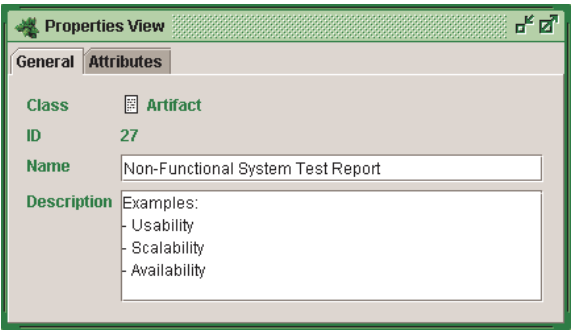


Figure 5.18 The General Properties tab of an entity

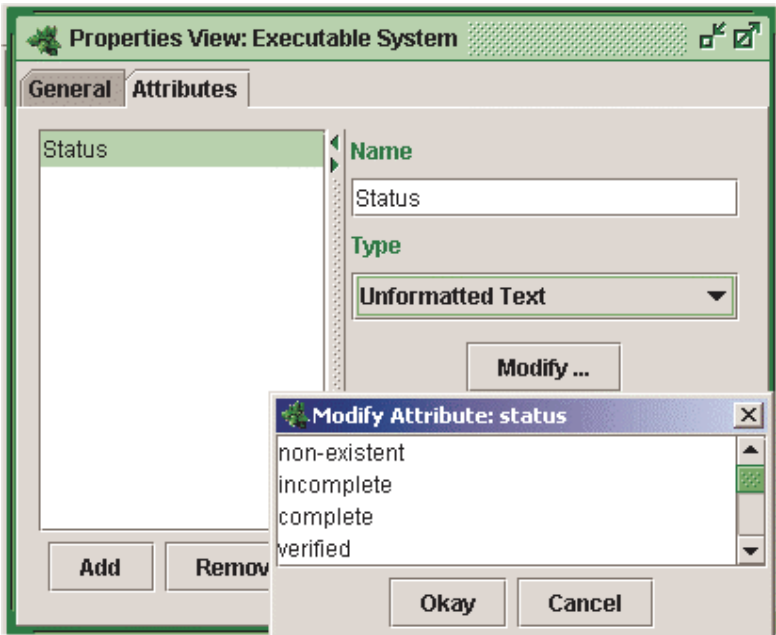


Figure 5.19 The Attributes tab for an entity

It is possible to resize the columns and the value and description text fields. Drag the split bars or use the small arrows to resize the components in this dialog.

Predefined sets of attributes can be activated via the attribute profiles option. For each entity, a name and a description can be specified. These are the default attributes for all entities in SPEARMINT™.

In the attributes tab (Fig 5.18), the user can define its own attributes, change the name of new or existing attributes, and define a value for the attribute.

Something to remember: The properties view is the mechanism that allows users to configure the process model information according to the specific needs of the organization. The attributes specified can be generated into process handbooks or EPG's (see Chapter 7).

Something to remember: use the properties view for entering additional information about process entities you will later on display in your process handbooks, such as tasks needed to perform an activity, links to examples or templates for artifacts, or required skills for roles. This information appears in the process handbooks generated from SPEARMINT™ process models.

6 Consistency Assurance

Process models should be *complete* and *correct* representations of reality. This presupposes that they are internally *consistent*. The interpretation of inconsistent models leads to ambiguities or contradictions. Bringing about and preserving consistency of process models consequently is an important task when creating and maintaining models. To ease this task for you, consistency assurance has been partly automated in SPEARMINT™.

The following sections first introduce the methodological prerequisites underlying consistency assurance in SPEARMINT™, namely the process metamodel and the consistency rules. Then, SPEARMINT™'s functions for automated consistency assurance are described.

6.1 Methodological Prerequisites

The SPEARMINT™ tool is designed to support a method for descriptive modelling of software development processes, which has been defined by Fraunhofer IESE. This method defines the concepts and notations to be used for process models, as well as the activities and steps to be taken for eliciting and capturing process information into such models.

The specific methodological prerequisites that are relevant to the SPEARMINT™ tool are the process metamodel and the consistency rules. Both are given in the following sections.

6.1.1 Process Metamodel

Figure 6.1 shows the process metamodel that is used in SPEARMINT™. The core classes are «Entity», «Activity», «Artifact», «Role», «Resource», and «Tool», with the core associations «contains», «consumes», «modifies», «produces», «involves», «uses», and «precedes». The «Attribute» class is added for flexibility, while the «ProductFlowGraph» and «ControlFlowGraph» classes and their associations are added for visual modeling.

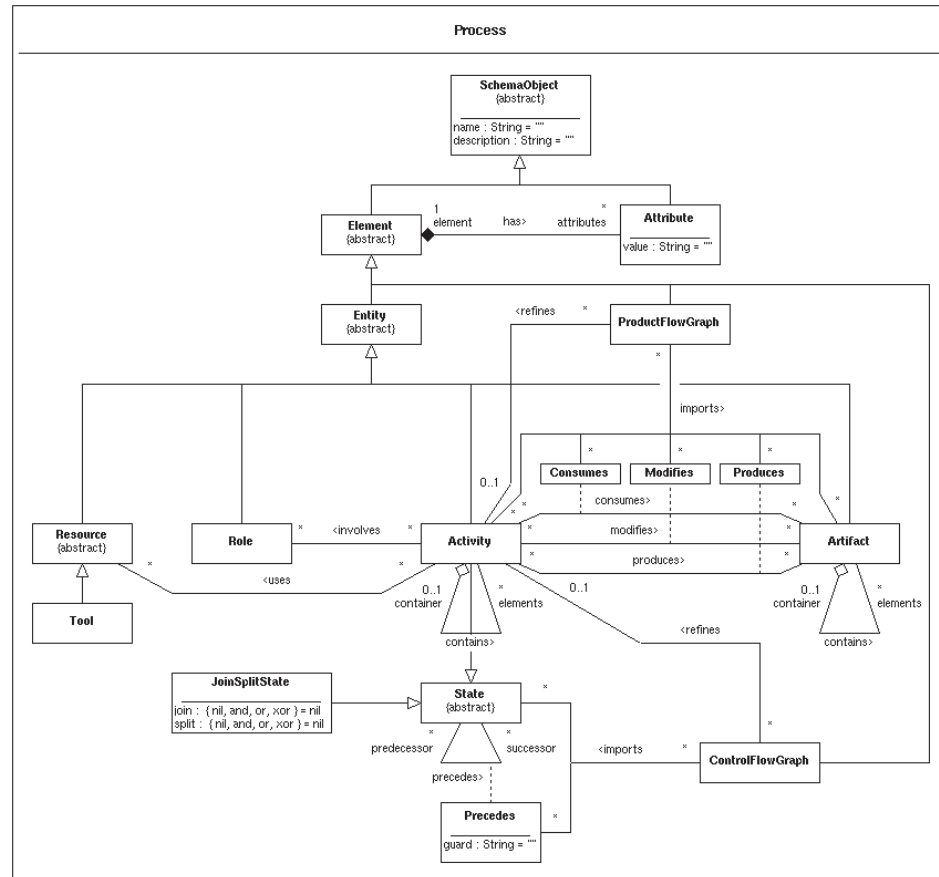


Figure 6.1 Process metamodel of SPEARMINT™

6.1.2 Consistency Rules

Every instance of class «Process» in the process metamodel is a process model that is required to conform to both the metamodel and the following set of consistency rules.

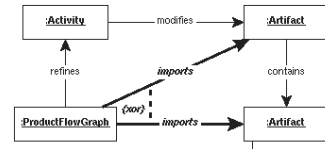
R1 An Attribute belongs to exactly one Element. (built-in¹)

¹ Rules R1, R6, R8, R12, and R37 are made explicit in the metamodel with multiplicity restrictions «1» or «0..1». In addition, rule R1 is related to a composition («has»), rules R6 and R8 are related to hierarchies («contains», cf. rules R9 and R10), and rules R12 and R37 are related to derived hierarchies (implied by «refines» and «imports»).

Visual modeling is expected to deal with composition and hierarchies in appropriate ways, e.g. with graphical user interface (GUI) components for lists and trees. Using these kinds of GUI components, violations of the above rules are considered very unintuitive for the user – and difficult to deal with by the programmer. Therefore, a decision was made to not allow violations of these rules. Consequently, they have been hard-coded (built-in) into the design of the software system.

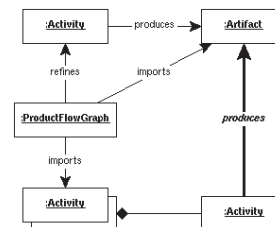
- R2 If a ProductFlowGraph imports a Consumes, Modifies, or Produces link, it must import the corresponding Activity and Artifact as well.
- R3 A ProductFlowGraph must import all Consumes, Modifies, and Produces links between imported Activities and Artifacts.
- R4 The name of an Attribute must be unique among the Attributes of the same Element.
- R5 Entities must have unique names.
- R6 An Activity is contained by at most one Activity. (built-in)
- R7 There must be exactly one root Activity. (A root Activity is not contained by an Activity.)
- R8 An Artifact is contained by at most one Artifact. (built-in)
- R9 An Activity must not contain itself, neither directly nor indirectly. (PF-4, first part)
- R10 An Artifact must not contain itself, neither directly nor indirectly. (PF-4, second part)
- R11 An Activity is refined by at most one ProductFlowGraph.
- R12 A ProductFlowGraph refines at most one Activity. (built-in)
- R13 There must be exactly one root ProductFlowGraph. (A root ProductFlowGraph does not refine an Activity.)
- R14 A root ProductFlowGraph imports exactly one Activity.
- R15 A root ProductFlowGraph imports only root Activities.
- R16 A root ProductFlowGraph imports only root Artifacts.
- R17 If a ProductFlowGraph refines an Activity, it imports exactly the element Activities of the refined Activity.
- R18 If a ProductFlowGraph refines an Activity and the refined Activity consumes, modifies, or produces an Artifact, the ProductFlowGraph imports either the Artifact or all of its element Artifacts.

Example scenario²:



- R19 A root Artifact must be imported by exactly one ProductFlowGraph.
- R20 An Activity must consume, modify, or produce at least one Artifact.
- R21 An Artifact must be consumed, modified, or produced by at least one Activity.
- R22 An Artifact is produced by at most one root Activity.
- R23 An Artifact must not be both consumed and produced, consumed and modified, or modified and produced by the same Activity.
- R24 If an Artifact is produced by an Activity and is imported by a ProductFlowGraph refining that Activity, the Artifact must be produced by exactly one Activity that is imported by the ProductFlowGraph.

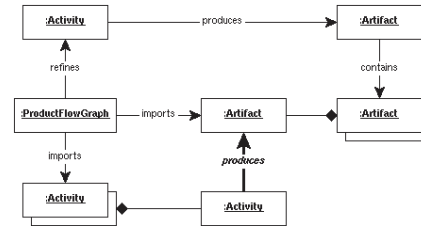
Example scenario:



- R25 If an Artifact is produced by an Activity and one of its element Artifacts is imported by a ProductFlowGraph refining that Activity, the element Artifact must be produced by exactly one Activity that is imported by the ProductFlowGraph.

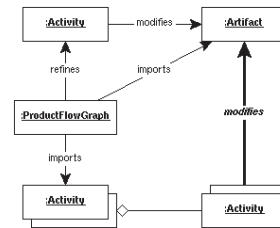
² Thin lines and non-italic fonts indicate the rules conditions, whereas thick lines and italic fonts indicate the rules implications.

Example scenario:



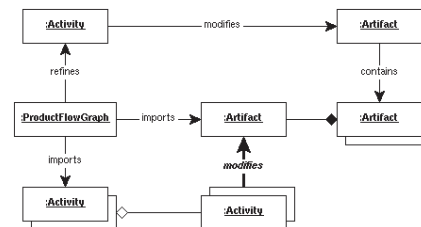
- R26 If an Artifact is modified by an Activity and is imported by a ProductFlowGraph refining that Activity, the Artifact must be modified by at least one Activity that is imported by the ProductFlowGraph.

Example scenario:



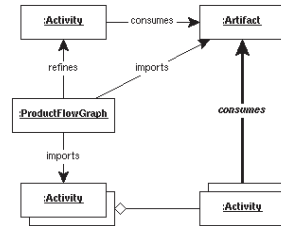
- R27 If an Artifact is modified by an Activity and one of its element Artifacts is imported by a ProductFlowGraph refining that Activity, the element Artifact must be modified by at least one Activity that is imported by the ProductFlowGraph.

Example scenario:



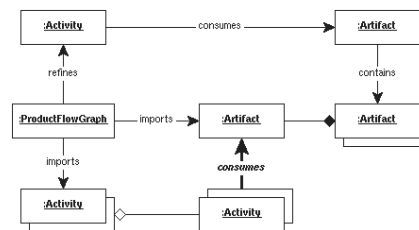
- R28 If an Artifact is consumed by an Activity and is imported by a ProductFlowGraph refining that Activity, the Artifact must be consumed by at least one Activity that is imported by the ProductFlowGraph.

Example scenario:



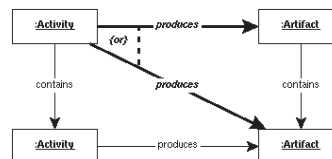
- R29 If an Artifact is consumed by an Activity and one of its element Artifacts is imported by a ProductFlowGraph refining that Activity, the element Artifact must be consumed by at least one Activity that is imported by the ProductFlowGraph.

Example scenario:



- R30 A non-root Activity may only produce an Artifact, if the Artifact or its container Artifact is produced by the container Activity.

Example scenario:



- R31 An Activity must be imported by exactly one ProductFlowGraph.
- R32 An Artifact must be imported by at least one ProductFlowGraph.
- R33 If a ProductFlowGraph imports an Artifact that is produced by an Activity, the ProductFlowGraph must also import the Activity; and vice versa..
- R34 If a ControlFlowGraph imports a Precedes link, it must import the corresponding States as well.
- R35 A ControlFlowGraph must import all Precedes links between imported States.

- R36 An Activity is refined by at most one ControlFlowGraph.
- R37 A ControlFlowGraph refines at most one Activity. (built-in)
- R38 There must be exactly one root ControlFlowGraph. (A root ControlFlowGraph does not refine an Activity.)
- R39 A root ControlFlowGraph imports exactly one Activity.
- R40 A root ControlFlowGraph imports only root Activities.
- R41 If a ControlFlowGraph refines an Activity, it imports exactly the element Activities of the refined Activity.
- R42 If a ControlFlowGraph imports a State, and the State succeeds or precedes a JoinSplitState, the ControlFlowGraph imports the JoinSplitState as well.
- R43 An Activity must be imported by exactly one ControlFlowGraph.
- R44 A JoinSplitState must be imported by exactly one ControlFlowGraph.
- R45 An Activity must precede and succeed at most one State.
- R46 If a JoinSplitState succeeds at most one State, its join must be nil; if it succeeds more than one State, its join must not be nil.
- R47 If a JoinSplitState precedes at most one State, its split must be nil; if it precedes more than one State, its split must not be nil.
- R48 A Precedes link may only have a non-empty guard, if the predecessor is a JoinSplitState with an `or` or `xor` split.
- R49 A Precedes link may only have a non-empty guard, if all other Precedes links to successors of the predecessor have a non-empty guard.
- R50 If an Activity produces an Artifact that is consumed or modified by another Activity, the producing Activity must precede the consuming or modifying Activity.
- R51 A State must not precede itself.
- R52 If an Activity contains at least one element activity, it is refined by at least one ProductFlowGraph.

R53 If an Activity contains at least one element activity, it is refined by at least one ControlFlowGraph.

6.2 Working with Automated Consistency Assurance

SPEARMINT™ provides three functions related to automated consistency assurance:

- Setting up automated consistency assurance
- Checking the consistency of a process model
- Showing and correcting inconsistencies on selected objects

These are described below.

6.2.1 Setting up Automated Consistency Assurance

Especially in early phases of modelling, it is necessary to be able to work with inconsistent models. Therefore, you can customize the automated consistency assurance to your personal needs. Open the consistency assurance dialog (Figure 6.2) by using the Tools menu and choosing the option »Consistency Assurance ...«.

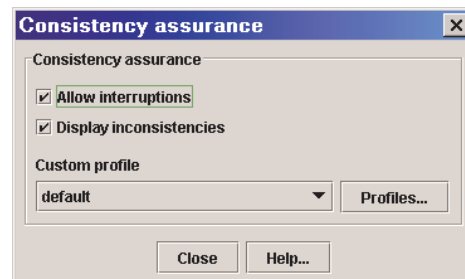


Figure 6.2: Consistency assurance dialog

- You can in general **allow** or **disallow interruptions** through dialogs that inform you about new inconsistencies.
- You can **switch on** and **off** the **displaying of inconsistencies** through colouring affected model elements red.
- You can **select** a defined **consistency assurance profile** for your further work with the model, or **define new profiles**.

Consistency assurance profiles determine the behaviour of the automated consistency assurance. Such a profile lays down for every single consistency rule how it is reacted on violations of this rule. On one hand, you can thereby customize the automated consistency assurance very flexibly to your needs. On

the other hand, you spare time in modelling since you can easily switch between the defined profiles.

Open the dialog »Custom profiles« (Figure 6.3) by clicking on the button »Profiles...«. With this dialog, you can **create** an arbitrary number of individual **profiles**, **configure** them, and eventually **delete** them. The standard profile »default« is invariable.

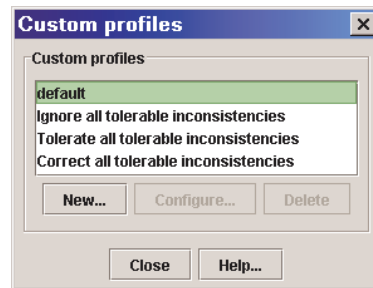


Figure 6.3: Custom profiles dialog

When you click on »New...« or »Configure...«, the dialog »Profile configuration« appears (Figure 6.4). With this dialog, you can configure the represented consistency assurance profile by laying down for every single consistency rule how it should be reacted on violations of this rule.

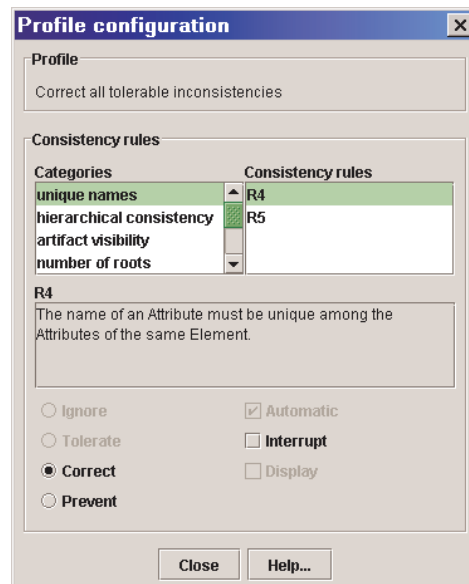


Figure 6.4: Profile configuration dialog

For clearness, the consistency rules have been assigned to various **categories**. Only the **consistency rules** that are assigned to the selected category are

listed. **Name** and **description** of the selected consistency rule are displayed. For laying down the **behaviour**, you have the following options:

- **Ignore** - Violations of the consistency rule are ignored completely. This does effectively switch off the rule.
- **Tolerate** - Violations of the consistency rule are detected as inconsistencies but tolerated. If you **switch off automatic** detection, you will be informed about inconsistencies on explicit request only. But you can also **switch on automatic** detection. Then you will eventually (**Interrupt on**) be informed about new inconsistencies through dialogs. In addition, inconsistencies are eventually (**Display on**) displayed through coloring affected model elements red.
- **Correct** - Violations of the consistency rule are automatically detected and corrected. You will eventually (**Interrupt on**) be informed about a new inconsistency and the upcoming correction through a dialog then.
Something to remember: There are some rules, which can't be automatically corrected at the moment. In these cases SPEARMINT behaves as the 'Prevent' option.
- **Prevent** - Operations, which would lead to violations of the consistency rule, are disabled. These operations are listed in menus and the like further on, but they appear gray and cannot be selected. Rationales for disabled operations are available as tool tips. So, if an operation appears gray in a menu, move the mouse over it and read the tool tip that appears to understand why.

Some consistency rules must not be violated on principle. In this case, only the options **Correct** and **Prevent** are available.

Interruptions through dialogs and displaying of inconsistencies through coloring affected model elements red can be switched off globally in the dialog »Consistency assurance«. The determinations made here are ineffective then.

6.2.2 Checking the Consistency of a Process Model

A consistency check can be performed on an entire process model, invoked from the process model popup menu in the database explorer. After checking, a window is displayed that shows the result of the consistency check (Figure 6.5).

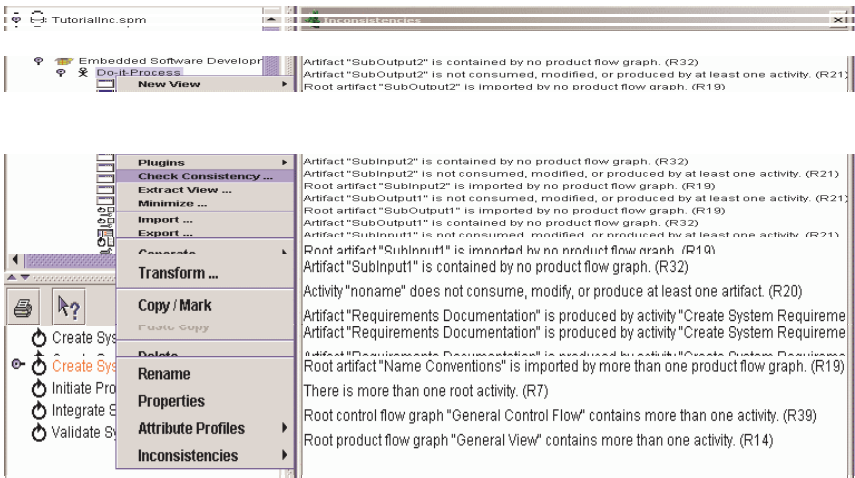


Figure 6.5 Inconsistencies

6.2.3 Showing and Correcting Inconsistencies on Selected Objects

The aim of the menu **Inconsistencies...** is to facilitate the elimination of inconsistencies on selected parts of the process model.

If an object (i.e., a view or a SPEARMINT™ entity) is selected and the mouse popup is invoked, all inconsistencies related to the currently selected object are listed in a short version.

New Sub-Artifact	
Copy / Mark	
Paste Copy	
Paste Object	
Delete	
Rename	
Inconsistencies	Activity "DefectPrevDetectionRate" prod>
	Activity "DefectPrevDetectionRate" prod>

Figure 6.6: Inconsistencies of an Entity

Moving your mouse pointer on such a list entry, the tool tip displays the long version of the error message. Selecting a list entry, SPEARMINT™ offers suggestions on how the inconsistency can be removed. The user can select one of these suggestions to correct the model accordingly.

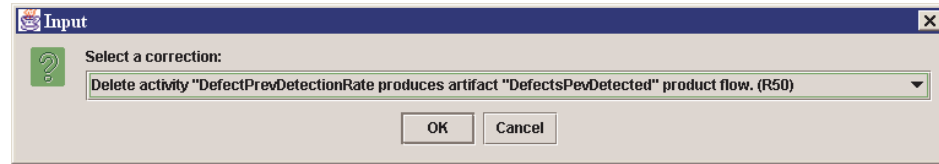


Figure 6.7: Solution suggestions for solving the selected inconsistency.

7 Similarity Analysis

7.1 Introduction

SPEARMINT™ supports the Multi-View Modeling (MVM) approach. Essential elements of this approach are role-specific modeling of the different development processes, and the integration of these models (*views*) in a *comprehensive model*.

Different points of view as well as different levels of abstraction applied by different roles originate mostly inconsistencies between the modeled views, which makes their integration difficult.

There are two kinds of inconsistencies between views:

- Different real world objects are modeled identically.
- Identical real world objects are modeled differently.

The main condition for discovering such inconsistencies is an effective similarity analysis applied to the entities modeled in the views. Therefore a tool for analyzing similarity between views supports the process engineer in integrating role-specific views in a comprehensive model.

7.1.1 Usage

Start the similarity analysis tool by clicking the right mouse button on a project, and by selecting the menu item »Similarity Analysis«.

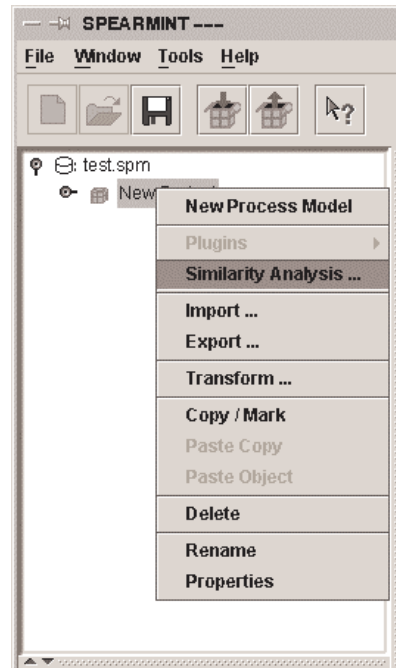


Figure 7.1: Starting similarity analysis.

A dialog is shown to select the process models to be analyzed.

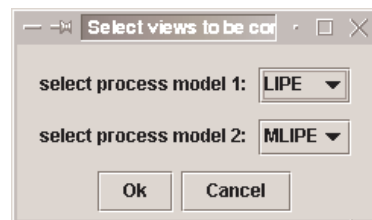


Figure 7.2: Selecting the process models to compare.

After the selection, all entities of both views are presented on four tables (*matching tables*), i.e. one table for each kind of SPEARMINT™ entity.

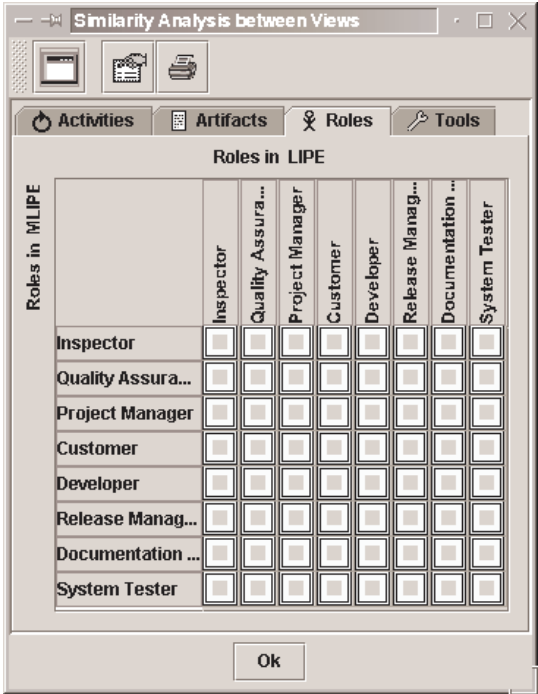


Figure 7.3: Matching tables.

All similarity values are initialized to 50%. This is a neutral value as 100% means equality and 0% means that two entities are completely different.

A click on the run button  starts a new similarity computation.

Similarity values computed by the tool are always *assumptions*: you can validate or refute them by converting them to positive or negative *facts*. A click on the right mouse button by pointing at the similarity value icon invokes an edit menu. An assumption can be set to a positive or a negative fact. An unsure fact can be reconverted into an assumption by clicking on »Reset«.

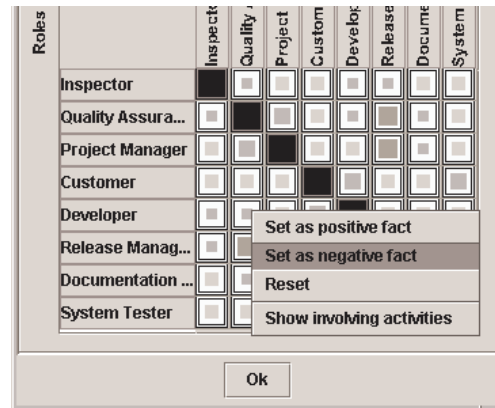


Figure 7.4: Setting a similarity fact.

A computation does not modify entered facts. Entered facts influence the whole computation.

Similarity icons represent similarity values as follows:












Icon	Semantic
	Similarity assumption, Value $\in \left[0, \frac{5}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{5}{100}, \frac{15}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{15}{100}, \frac{25}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{25}{100}, \frac{35}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{35}{100}, \frac{45}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{45}{100}, \frac{55}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{55}{100}, \frac{65}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{65}{100}, \frac{75}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{75}{100}, \frac{85}{100}\right]$
	Similarity assumption, Value $\in \left(\frac{85}{100}, \frac{95}{100}\right]$
	Similarity assumption, Value $\in \left[\frac{95}{100}, 1\right]$

Figure 7.5: Icons representing similarity values.



Icon	Semantic
	Similarity fact, Value = 0
	Similarity fact, Value = 1

Figure 7.6: Icons representing similarity facts.

Since matching tables may become very large sub-tables are also provided. To activate a sub-table choose two entities by clicking the right mouse button on a similarity value and select one of the »show« items.

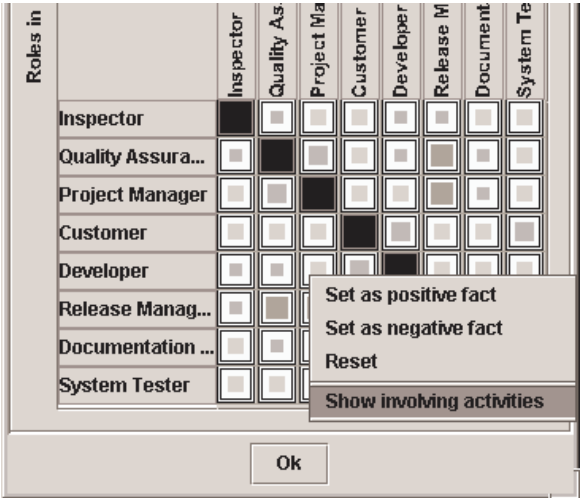


Figure 7.7: Showing a sub-table.

A sub-table is shown containing only entities, that relate to the chosen entities.

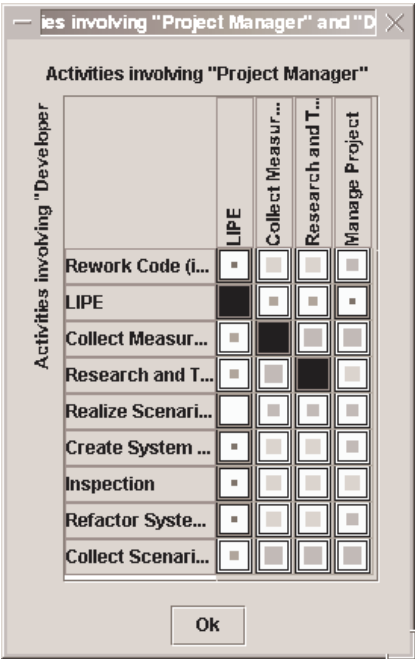


Figure 7.8: A sub-table.

Different kinds of sub-tables can be generated depending on the type of the chosen entities.

		Matching Tables			
		Activities	Artifacts	Roles	Tools
Related Sub-Tables	Activities	Contained sub-activities	Using activities	Involving activities	Using activities
	Artifacts	Accessed artifacts	Contained sub-artifacts		
	Roles	Involved roles			
	Tools	Used tools			

The similarity values within a sub-table are editable too (right mouse button). Therefore, sub-tables provide comfortable detailed views of the model.

7.2 Heuristics

The similarity computation rests upon several heuristics, which can be classified into the following five groups.

Heuristics for all kind of entities	Heuristics for artifacts	Heuristics for roles	Heuristics for tools	Heuristics for activities
Name	PdAgg	RsAggRole	RsAggTool	PcAgg
Attr				SC
ASt				PdS
				PdF
				PcH
				PcS
				RsT
				PcM

The name heuristic (**Name**) computes the similarity between entity identifiers by the weighted Levenshtein distance. The computation applies the formula:

$$Name(a, b) = \left(\frac{WLSD(name_a, name_b)}{\max(length(name_a), length(name_b))} \right),$$

where $WLSD(name_a, name_b)$ is the Levenshtein distance between the names of two models a and b, $\max(length(name_a), length(name_b))$ is the greatest of both name lengths.

The aggregation structure heuristic (**ASt**) computes the similarity between two models by their aggregation degree: two entities are equal if and only if both are aggregations or not.

The attribute heuristic (**Attr**) computes the similarity between two models by the similarity between their attribute lists. The following formula is applied:

$$Attr(a, b) = \begin{cases} 0, & \text{if } (((list_a = \emptyset) \text{ and } (list_b \neq \emptyset)) \text{ or } ((list_a \neq \emptyset) \text{ and } (list_b = \emptyset))) \\ 1, & \text{if } (((list_a = \emptyset) \text{ and } (list_b = \emptyset)) \text{ or } ((list_a \neq \emptyset) \text{ and } (list_b \neq \emptyset))) \\ & \sigma(list_a, list_b), \text{ else.} \end{cases}$$

Here, the similarity measure σ between two non-empty lists of attributes $l_1 = (x_1, \dots, x_n)$, $l_2 = (y_1, \dots, y_m)$ is

$$\sigma(l_1, l_2) = \max \left(\left(\frac{\sum_{i=1}^n \max(sim(x_i, y_1), \dots, sim(x_i, y_m))}{n} \right), \left(\frac{\sum_{j=1}^m \max(sim(x_1, y_j), \dots, sim(x_n, y_j))}{m} \right) \right)$$

The similarity measure between single attributes $sim(x, y)$ is computed by a weighted average of the attribute name similarity, the attribute value similarity and the attribute type similarity:

$$sim(x, y) = w \cdot Name(x, y) + y \cdot equal(type_x, type_y) + z \cdot equal(value_x, value_y)$$

where w = Attribute Name Amount, y = Attribute Type Amount, z = Attribute Value Amount.

The product aggregation heuristic (**PdAgg**) can be applied only to artifacts. Its results are represented in the artifact matching table. PdAgg computes the similarity between artifacts by a weighted average of the Name, Attr, and ASt similarity measures:

$$PdAgg(a, b) = w \cdot Name(a, b) + y \cdot ASt(a, b) + z \cdot Attr(a, b)$$

where w = Artifact Name Amount, y = Artifact ASt Amount, z = Artifact Attr Amount.

The role aggregation heuristic (**RsAggRole**) can be applied only to roles. Its results are represented in the role matching table. RsAggRole computes the similarity between roles by a weighted average of the Name and the Attr similarity measures:

$$RsAggRole(a, b) = w \cdot Name(a, b) + y \cdot Attr(a, b)$$

where w = Role Name Amount, y = Role Attr Amount.

The tool aggregation heuristic (**RsAggTool**) can be applied only to tools. Its results are represented in the tool matching table. RsAggTool computes the similarity between tools by a weighted average of the Name and the Attr similarity measures:

$$RsAggTool(a, b) = w \cdot Name(a, b) + y \cdot Attr(a, b)$$

where w = Tool Name Amount, y = Tool Attr Amount.

The process aggregation heuristic (**PcAgg**) is applied to activities. PcAgg computes the similarity between activities by a weighted average of the Name and the Attr similarity measures:

$$PcAgg(a, b) = w \cdot Name(a, b) + y \cdot Attr(a, b)$$

where w = Activity Name Amount, y = Activity Attr Amount.

The structure compatibility heuristic (**SC**) is applied to lists of artifacts or activities related to the two activities to be analyzed and measures the capability to map the elements of one list into the elements of the other list. Let $\text{sim}(a, b)$ be a similarity function between two list of artifacts or activities

$l_a = \{a_1, \dots, a_n\}$, $l_b = \{b_1, \dots, b_m\}$ and $\text{isFact}(\text{sim}(a, b))$ a predicate, true if $\text{sim}(a, b)$ is a fact. Furthermore $p \in \{0, 1, 2, \dots\}$, $p \leq n$ is the number of elements from l_a with the following property: There is at least one element $b \in l_b$ with $\text{sim}(a, b) = 1.0$ and $\text{isFact}(\text{sim}(a, b))$, i.e.,

$$p = \left| \left\{ a \in l_a : (\exists b \in l_b : \text{sim}(a, b) = 1.0, \text{isFact}(\text{sim}(a, b))) \right\} \right|.$$

Accordingly is $q \in \{0, 1, 2, \dots\}$, $q \leq m$ the number of elements from l_b with $\text{sim}(a, b) = 1.0$ and $\text{isFact}(\text{sim}(a, b))$, i.e.,

$$q = \left| \left\{ b \in l_b : (\exists a \in l_a : \text{sim}(a, b) = 1.0, \text{isFact}(\text{sim}(a, b))) \right\} \right|.$$

The function $SC(l_a, l_b)$ is defined as follows:

$$SC(l_a, l_b) = \frac{2 \cdot \min(p, q)}{n + m}.$$

Since the SC heuristic works only with positive facts the parameter Threshold determines the minimal similarity value for an assumption to be considered a positive fact.

The product structure heuristic (**PdS**) computes the similarity between two activities resting upon the PdAgg similarity measure of their artifacts. Given an activity a its artifact interface is defined as $c(a) \cup m(a) \cup p(a)$, where $c(a)$ is the set of artifacts consumed by a , $m(a)$ the set of modified artifacts, and $p(a)$ are the artifacts produced by a . Therefore the artifact interface of a is the set of all artifacts accessed by a . Let $t \in [0.0, 1.0]$ be a similarity threshold value, and for activities a, b , $T_{SC(a,b)}$ the sub matching table of PdAgg concerning their artifact interfaces:

$$T_{SC(a,b)}[i, j] = \text{PdAgg}(x_i, y_j), x_i \in c(a) \cup m(a) \cup p(a), y_j \in c(b) \cup m(b) \cup p(b).$$

The PdS similarity measure is computed by setting positive facts in $T_{SC(a,b)}[i, j]$ and applying the SC heuristic to it. The positive facts are automatically set with the following strategy:

1. Convert each similarity assumption $\text{sim}(x, y)$ greater then the SC parameter Threshold into a positive fact, if:
 - $x \in c(a)$ and $y \in c(b)$ or
 - $x \in m(a)$ and $y \in m(b)$ or
 - $x \in p(a)$ and $y \in p(b)$,

i.e., if the concerning artifacts are both consumed, modified, or produced.

2. If there are neither facts concerning artifact x , nor facts concerning artifact y , convert each similarity assumption $\text{sim}(x, y)$ greater then the SC parameter Threshold into a positive fact, if:

$x \in c(a)$ and $y \in m(b)$ or

$x \in m(a)$ and $y \in c(b)$ or

$x \in p(a)$ and $y \in m(b)$ or

$x \in m(a)$ and $y \in p(b)$.

Note that these artifacts are accessed by the activities a and b in a similar way.

3. If there are neither facts concerning artifact x , nor facts concerning artifact y , convert each similarity assumption $\text{sim}(x, y)$ greater then the SC parameter Threshold into a positive fact, if:

$x \in c(a)$ and $y \in p(b)$ or

$x \in p(a)$ and $y \in c(b)$,

Note that these artifacts are accessed by the activities a and b in a different way.

The product flow heuristic (**PdF**) computes the similarity between two activities resting upon the PdAgg similarity measure of their artifacts. Like the PdS heuristic, the PdF transforms similarity assumptions in the sub matching table $T_{SC(a,b)}[i, j]$ concerning the artifacts accessed by the activities a and b into positive facts and applies the SC heuristic to compute the similarity values. It works with the following strategy:

1. Convert each similarity assumption greater then the SC parameter Threshold into a positive fact.
2. $PdF(a, b) = \max(PdF1(a, b), PdF2(a, b), PdF3(a, b))$, where

$$PdF1(a, b) = w_{cc} \cdot SC(c(a), c(b)) + w_{mm} \cdot SC(m(a), m(b)) + w_{pp} \cdot SC(p(a), p(b))$$

$$PdF2(a, b) = w_{cm} \cdot SC(c(a), m(b)) + w_{mc} \cdot SC(m(a), c(b)) + w_{pm} \cdot SC(p(a), m(b)) + w_{mp} \cdot SC(m(a), p(b))$$

$$PdF3(a, b) = w_{cp} \cdot SC(c(a), p(b)) + w_{pc} \cdot SC(p(a), c(b))$$
 where $w_{xy} = x-y$ Amount in the PdF heuristic family.

The process hierarchy heuristic (**PcH**) computes a similarity values between two activities resting upon the PcAgg similarity of their sub-activities. Since a comparison of the whole aggregation trees can become very computational complex, the computation is only concerned with the first three hierarchy levels of the tree structure. Since different aggregation tree structures may imply the same semantics, different aggregation levels are compared independently of the tree structure, i.e., the trees are converted into one-level lists.

Given the lists of sub-activities of a and b, $l_a = (x_1, \dots, x_n)$ and $l_b = (y_1, \dots, y_m)$, the following formula is applied:

$$PcH(a, b) = \begin{cases} 0, & \text{if } ((l_a = \emptyset) \text{ and } (l_b \neq \emptyset)) \text{ or } ((l_a \neq \emptyset) \text{ and } (l_b = \emptyset)) \\ 1, & \text{if } ((l_a = \emptyset) \text{ and } (l_b = \emptyset)) \text{ or } ((l_a \neq \emptyset) \text{ and } (l_b \neq \emptyset)) \\ \sigma(l_a, l_b), & \text{else.} \end{cases}$$

Here, the similarity measure σ between two non-empty lists of sub-activities is

$$\sigma(l_a, l_b) = \max \left(\left(\frac{\sum_{i=1}^n \max(PcAgg(x_i, y_1), \dots, PcAgg(x_i, y_m))}{n} \right), \left(\frac{\sum_{j=1}^m \max(PcAgg(x_1, y_j), \dots, PcAgg(x_n, y_j))}{m} \right) \right)$$

The process structure heuristic (**PcS**) computes, like the PcH, a similarity values between two activities resting upon the PcAgg similarity of their sub-activities. Like in PcH the sub-activities of two activities a and b belonging the first three level of the aggregation trees are collected in two lists $l_a = (x_1, \dots, x_n)$ and $l_b = (y_1, \dots, y_m)$. Given the sub-matching table of PcAgg related to l_a and l_b , the computation is performed with a similar strategy like in the PdF heuristic:

1. Convert each similarity assumption greater then the SC parameter Threshold into a positive fact.
2. $PcS(a, b) = SC(l_a, l_b)$.

The resource type heuristic (**RsT**) computes a similarity value between two activities a and b using similarity results of the Name heuristic applied to their related roles and tools. It is defined as:

$$RsT(a, b) = x \cdot \sigma(roles_a, roles_b) + y \cdot \sigma(tools_a, tools_b),$$

where x = Role Name Amount, y = Tool Name Amount, and

$$\sigma(l_a, l_b) = \max \left(\left(\frac{\sum_{i=1}^n \max(Name(x_i, y_1), \dots, Name(x_i, y_m))}{n} \right), \left(\frac{\sum_{j=1}^m \max(Name(x_1, y_j), \dots, Name(x_n, y_j))}{m} \right) \right)$$

The process model heuristic (**PcM**) compounds all heuristic for activities. Its results are presented in the activity matching table. For two given activities a, b $PcM(a, b)$ is defined as follows:

$$PcM(a, b) = w_{PdS} \cdot PdS(a, b) + w_{PdF} \cdot PdF(a, b) + w_{PcH} \cdot PcH(a, b) + w_{PcS} \cdot PcS(a, b) + w_{RsT} \cdot RsT(a, b)$$

Here is w_x = x Amount.

The relationships among the different heuristics are displayed by the following diagram:

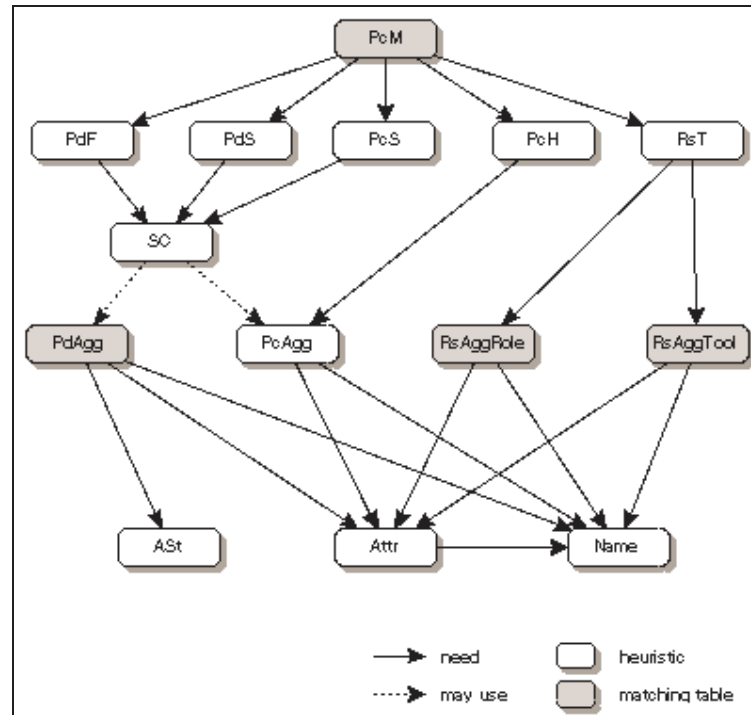


Figure 7.9: Hierarchy of similarity heuristics.

7.3 Parameter setting

There are several parameters influencing the behavior of the different heuristics. A click on the setting button  invokes a parameter edit dialog.

All parameters are grouped in the dialog by their related heuristic. As the parameters determine the relevancy of the subordinated heuristics within the heuristic they belong to, their total amount is always 100%, i.e. if you edit a parameter the tool will adjust the other parameters automatically.

The parameter value 0% switches off the related heuristic, whereas all other involved heuristics are switched off by the value 100%.

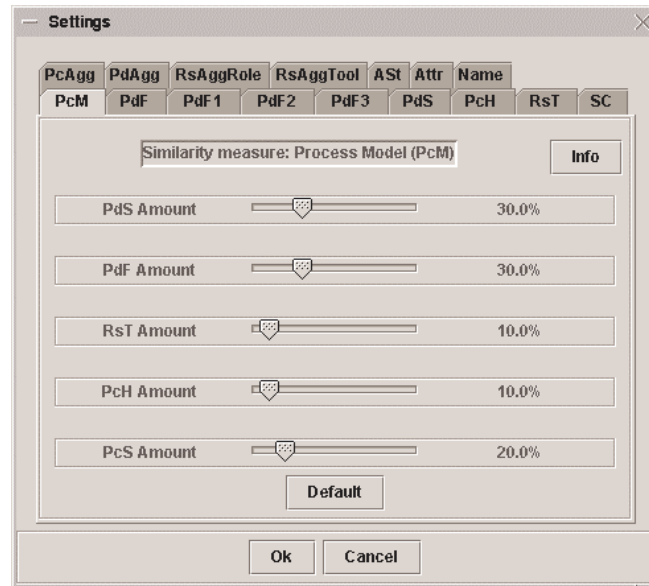


Figure 7.10: Setting parameters of similarity heuristics.

The parameter »Threshold« of the SC heuristic does not determine the relevancy of any heuristic, therefore it can be different than 100%.

You can find a description of each parameter in section 7.2.

8 Electronic Process Guide (EPG) Generation

With the EPG generator you can generate an HTML based EPG of the process model in SPEARMINT™. The EPG is a web-based description of the process. It is a set of HTML pages that are browseable in any web browser.

Currently there are two options available for this purpose, which are:

- **Electronic Process Guide:** This option generates an electronic process guide, based on a fix template, which is not visible to the user.
- **SPEARMINT™ 5 EPG:** This option allows the generation of a SPEARMINT™ 5 Electronic Process Guide.

8.1 SPEARMINT™ 7 EPG generation:

The steps to generate are:

- Right click on the process model
- Select the option **Generate Electronic Process Guide...**
- Select the directory where the documentation will be placed
- A dialog box warning that all files and subdirectories will be deleted appears. Verify once again the directory. Click ok.
- A new dialog pops up during the generation until it is completed.

In the directory selected previously for generation, open the browser and select the file index.html.

8.2 SPEARMINT™ 5 EPG: Setting the Configuration

8.2.1 Changing the Settings for Graphical Views

Copy the document 'DefaultConfiguration.xml', which lies in the 'lib' file of the SPEARMINT™ folder, to a personal directory. Rename it 'PersonalConfiguration.xml':

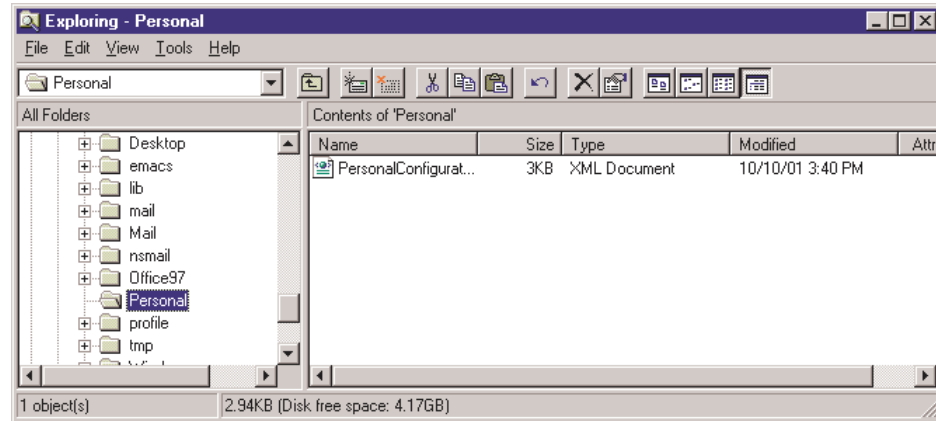


Figure 8.1 The renamed copy of the default configuration document.

Open the document 'PersonalConfiguration.xml' in the windows notepad and change the configuration in the following line:

```
<Configuration dtd="0.1">
    <Property key="epg.graphicalViews">true</Property>
```

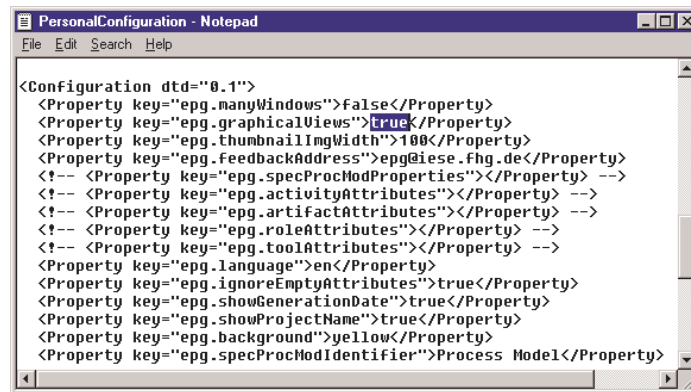


Figure 8.2 PersonalConfiguration – Windows notepad

The configuration of the EPG can be set in a configuration file (see section 3.7.1).

```
<Property key="epg.manyWindows">false</Property>
```

- The value **true** will open a new window in the web browser for every new inquiry.
- The value **false** will open every inquiry in the same window of the web browser.

```
<Property key="epg.graphicalViews">true</Property>
```

- Determines whether the EPG will include graphical views, possible values are therefore true or false.

<Property key="epg.thumbnailImgWidth">**100**</Property>

- Determines the width of the thumbnail image for the graphical views.

<Property key="epg.feedbackAddress">**epg@iese.fhg.de**</Property>

- Gives the email feedback address to be displayed in the EPG.

<!-- <Property key="epg.artifactAttributes">status</Property> -->

- **<!-- -->** this line is deactivated in the configuration; to activate the line, delete the outer brackets
- Determines which attributes are to be displayed in the EPG. The attributes have to be named explicitly (case sensitive!) and if there are more than one attribute, they have to be divided by three # symbols, i.e.:
 <Property key="epg.artifactAttributes">**status###size**</Property>
- If the property is deactivated, all attributes will be displayed in the EPG.

<Property key="epg.language">**en**</Property>

- Determines the language of the EPG; to change it to German, the following change has to be made:

 <Property key="epg.language">**de**</Property>

<Property key="epg.ignoreEmptyAttributes">**true**</Property>

- Determines whether the EPG will display empty attribute, possible values are true or false.

<Property key="epg.showGenerationDate">**true**</Property>

- Determines whether the date when the EPG has been generated shall be displayed, possible values are true or false.

<Property key="epg.showProjectName">**true**</Property>

- Determines whether the name of the project is to be displayed, possible values are true or false.

<Property key="epg.background">**white**</Property>

- Determines the background color of the EPG in the browser window.

<Property key="epg.artifactsIdentifier">**Artifacts**</Property>

- A given entity is identified within a certain heading. All artifacts could be documents, for example. In order for them to be labeled as such in the EPG, the configuration would have to be changed as follows:
<Property key="epg.artifactsIdentifier">**Documents**</Property>

8.2.2 Generating a SPEARMINT™ 5 EPG

Choose the menu item **Generate EPG ...** in the popup menu of a process model.

A file-choose dialog opens, where the root folder for the EPG has to be selected. Select the folder but do not enter it, otherwise the generator will not start.

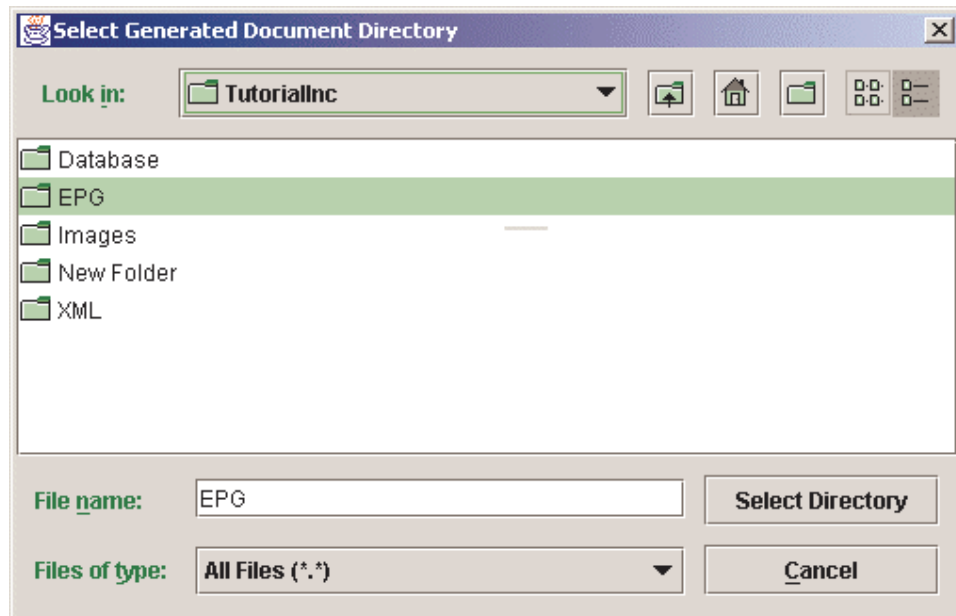


Fig 8.3

Generate an EPG

In the next file-choose dialog, the configuration file for the EPG can be selected. If no file is selected, the generator takes the default configuration file (see Fig 7.2).

For defining an individual configuration file, copy the default file (DefaultConfiguration.xml) and change it to suit your own needs as described in section 7.1.

Now generation will start. The status of the generation is shown in the status bar.

8.2.3 Open a SPEARMINT™ 5 EPG

To open the EPG, load the file Index.html in the folder EPG into your web browser.

The EPG uses java applets and is optimized for the Netscape Navigator.

8.3 Structure of the EPG

The HTML based EPG's divides the browser window into three frames, the two left frames are for navigation and the right frame is for displaying information on the process.

The top frame of the two left frames displays a tree with all entities of the process; activities, artifacts, roles, and tools.

The lower left frame contains a dynamic table of contents, which changes according to the selection made in the upper frame.

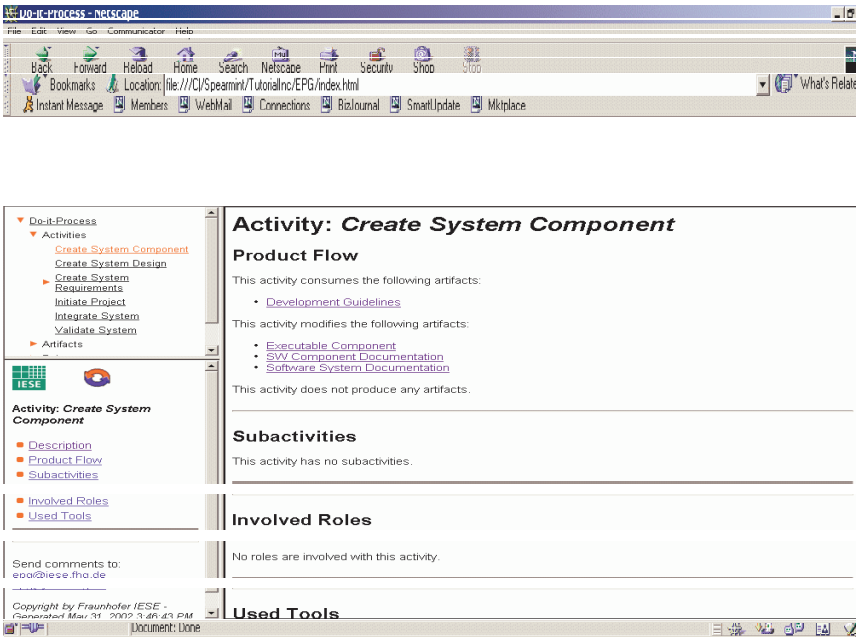


Fig 8.4 SPEARMINT™ 7's EPG

9 Plugins

Plugins are small programs that access the SPEARMINT™ database through a published API (application programming interface). Some plugins are provided with SPEARMINT™; others may be written by third parties.

9.1 Calculate Basis Statistics

The **Calculate Basis Statistics** plugin basically counts the elements of the object it is invoked on (database, project, or process model).

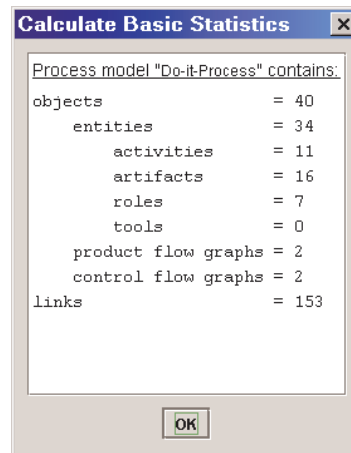


Fig 9.1 Calculate Basis Statistics

9.2 Calculate Size and Complexity

The **Calculate Size and Complexity** plugin calculates size and complexity measures for a process model.

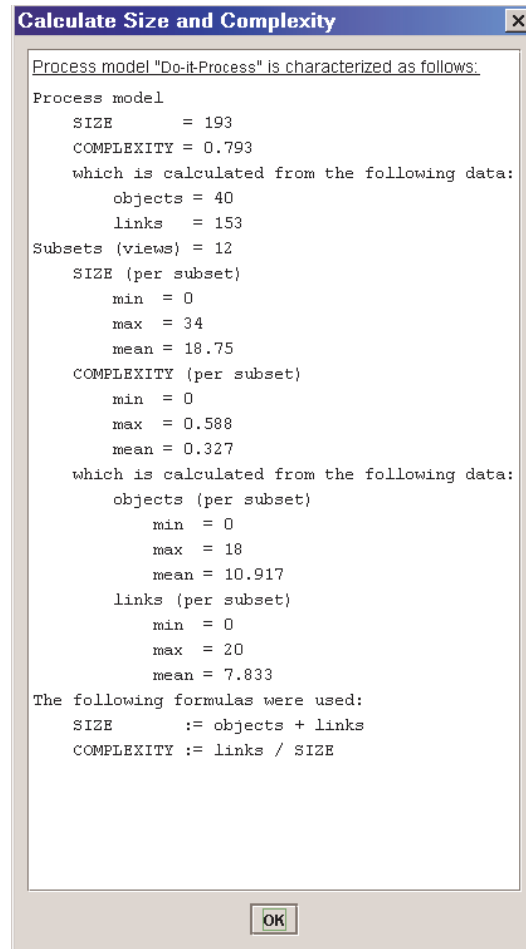


Fig 9.2

Calculate Size and Complexity

9.3 Find Undefined Attributes

The **Find Undefined Attributes** plugin searches entities in a process model, that have attributes with an empty value. These entities and their empty attributes are displayed in a result list. The user can click on any entry in this list to directly focus the properties view on the corresponding attribute.

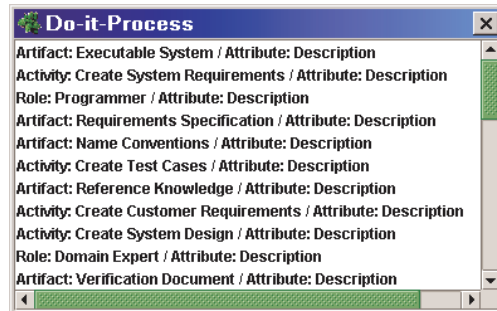


Fig 9.3 Find Undefined Attributes

9.4 Outdated Plugins

There are some **Outdated Plugins** as well. These are prototypical implementations of research ideas. No support is provided for them.

9.5 Writing Plugins

To design and implement plugins for SPEARMINT™, providers should have a look at the SPEARMINT™ API documentation:

```
<SPEARMINT™ installation directory>/doc/api/index.html
```

SPEARMINT™ plugins have to implement the following interface:

```
spearmint.plugin.RevisedSpearmintPlugin
```

For compilation, the following libraries have to be put in the CLASSPATH environment variable:

```
<SPEARMINT™ installation directory>/lib/spearmint.jar
<SPEARMINT™ installation directory>/lib/framework.jar
<SPEARMINT™ installation directory>/lib/omf.jar
```

For activation, plugins must be put in the following directory:

```
<SPEARMINT™ installation directory>/plugins/
```

To activate a plugin, it has to be added to the property

```
spearmint.revisedplugins
```

in the user's file

```
$(HOME)/.spearmint
```

This property contains a list of comma-separated Java class names.

Document Information

Title: SPEARMINT™ 7:
User Manual

Date: January, 2003
Report: IESE-072.02/E
Status: Final
Distribution: Public

Copyright 2003, Fraunhofer IESE.
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means including, without limitation, photocopying, recording, or otherwise, without the prior written permission of the publisher. Written permission is not needed if this publication is distributed for non-commercial purposes.